

Quick Search

Advanced Search

Number Search

Last result list

My patents list

Classification Search

Can be searched

- Quick Help
- » Why are some tabs deactivated for certain documents?
  - » Why does a list of documents with the heading "Also published as" sometimes appear, and what are these documents?
  - » What does A1, A2, A3 and B stand for after an EP publication number in the "Also published as" list?
  - » What is a cited document?
  - » What are citing documents?
  - » What information will I find if I click on the link "View all"?
  - » What information will I find if I click on the link "View document in the European Register"?
  - » Why do I sometimes find the abstract of a corresponding document?
  - » Why isn't the abstract available for XP documents?

Object oriented video system

Bibliographic data

Description

Claims

Original document

INPADOC legal status

Publication number: CN1402852 (A)

Publication date: 2003-03-12

Inventor(s): GONZALEZ RUBEN [US] +

Applicant(s): ACTIVESKY INC [US] +

Classification:

- international: G06F17/30; G06T13/00; G11B27/10; H04L12/28; H04L12/56; H04L29/06; H04N11/04; H04N5/92; H04N7/08; H04N7/081; H04N7/173; H04N7/24; H04N7/26; H04N7/28; H04N7/52; H04N7/66; H04N7/16; G06F17/30; G06T13/00; G11B27/10; H04L12/28; H04L12/56; H04L29/06; H04N11/04; H04N5/92; H04N7/08; H04N7/081; H04N7/173; H04N7/24; H04N7/26; H04N7/52; H04N7/64; H04N7/16; (IPC1-7): G06F17/30; H04L12/56; H04N7/26

- European: G06F17/30E; G06F17/30S8T; G11B27/10; H04L12/28W; H04L12/56B; H04L29/06C2; H04L29/06M6C2; H04L29/06M6E; H04N7/24C12C; H04N7/26A6C8; H04N7/26J12; H04N7/26J8; H04N7/26J8; H04N7/26Z10; H04N7/28; H04N7/52; H04N7/66

Application number: CN20008016364 20001020

Priority number(s): AU1999PQ03603 19991022; AU2000PQ08661 20000707

View INPADOC patent family

View list of citing documents

Also published as:

WO0131497 (A1)

US2007005795 (A1)

TW229559 (B)

NZ518774 (A)

MXPA02004015 (A)

more >>

Report a data error here

Abstract not available for CN 1402852 (A)

Abstract of corresponding document: **WO 0131497 (A1)**

Translate this text

A method of generating an object oriented interactive multimedia file, including encoding data comprising at least one of video, text, audio, music and/or graphics elements as a video packet stream, text packet stream, audio packet stream, music packet stream and/or graphics packet stream respectively, combining the packet streams into a single self-contained object, said object containing its own control information, placing a plurality of the objects in a data stream, and grouping one or more of the data streams in a single contiguous self-contained scene, the scene including format definition as the initial packet in a sequence of packets. An encoder for executing the method is provided together with a player or decoder for parsing and decoding the file, which can be wirelessly streamed to a portable computer device, such as a mobile phone or a PDA. The object controls provide rendering and interactive controls for objects allowing users to control dynamic media composition, such as dictating the shape and content of interleaved video objects, and control the objects received.



Data supplied from the **espacenet** database — Worldwide

http://v3.espacenet.com/publicationDetails/biblio?DB=EPODOC&adjacent=true&locale=en\_EP&FT=D&date=20030312&CC=CN&NR=1402852A&KC=A[7/26/2010 11:37:26 AM]

[19] 中华人民共和国国家知识产权局

[ 51 ] Int. Cl<sup>7</sup>

G06F 17/30

H04L 12/56 H04N 7/26



[12] 发明专利申请公开说明书

[21] 申请号 00816364.2

[43] 公开日 2003 年 3 月 12 日

[11] 公开号 CN 1402852A

[22] 申请日 2000.10.20 [21] 申请号 00816364.2  
[30] 优先权  
[32] 1999.10.22 [33] AU [31] PQ3603  
[32] 2000. 7. 7 [33] AU [31] PQ8661  
[86] 国际申请 PCT/AU00/01296 2000.10.20  
[87] 国际公布 WO01/31497 英 2001.5.3  
[85] 进入国家阶段日期 2002.5.28  
[71] 申请人 动感天空公司  
地址 美国加利福尼亚  
[72] 发明人 鲁宾·冈萨雷斯

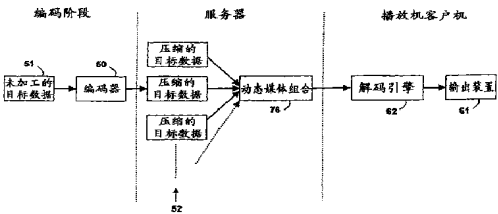
[74] 专利代理机构 中国国际贸易促进委员会专利  
商标事务所  
代理人 王以平

权利要求书 29 页 说明书 131 页 附图 46 页

[54] 发明名称 面向目标的视频系统

[57] 摘要

一种产生面向目标交互多媒体文件的方法，包括：编码至少由视频、文本、音频、音乐和/或图形单元之一组成的数据分别作为视频包流、文本包流、音频包流、音乐包流和/或图形包流；组合所述包流为单一自包含的目标，所述目标含有其自己的控制信息；放置多个所述目标到数据流中；以及在单一邻近的自包含场景中，分组一个或多个所述数据流，所述场景包括作为包序列中的初始包的格式定义。设有一个用于执行所述方法的编码器以及用于分析和解码所述文件的播放机或解码器，可无线地发送码流到诸如移动电话或 PDA 的便携计算机装置。目标控制提供目标的修饰和交互控制，允许用户控制动态媒体合成，如指示交叉视频目标的构形和内容并控制接收的目标。



1. 一种产生面向目标交互多媒体文件的方法，包括：

编码至少由视频、文本、音频、音乐和/或图形单元之一组成的数据分别作为视频包流、文本包流、音频包流、音乐包流和/或图形包流；

组合所述包流为单一自包含的目标，所述目标含有其自己的控制信息；

放置多个所述目标到数据流中；以及

在单一邻近的自包含场景中，分组一个或多个所述数据流，所述场景包括作为包序列中的初始包的格式定义。

2. 按照权利要求 1 的产生交互多媒体文件的方法，包括组合一个或多个所述场景。

3. 按照权利要求 1 的产生交互多媒体文件的方法，其中单一场景含有目标库。

4. 按照权利要求 1 的产生交互多媒体文件的方法，其中用于组成可订制的解压变换的数据包含在所述目标中。

5. 按照权利要求 1 的产生交互多媒体文件的方法，其中目标控制数据被附加到交叉到视频比特流中的各个目标上，并且所述目标控制数据控制交互行为、修饰参数、组合、以及压缩数据的翻译。

6. 按照权利要求 1 的产生交互多媒体文件的方法，包括分层目录结构，其中包含场景信息第一级别目录数据包括有所述第一场景，包含流信息的第二级别目录数据包括有一个或多个所述场景，并且其中包含识别帧内位置信息的第三级别目录数据包括在所述数据流中。

7. 一种产生面向目标交互多媒体文件的方法，包括：

编码至少由视频和音频单元之一组成的数据分别作为视频包流和音频包流；

组合所述包流为单一自包含组态目标；

放置所述目标在数据流中；

放置所述流到单一邻近的自包含场景中，所述场景包括格式定义；

和

组合多个所述场景。

8. 按照权利要求 1 的产生交互多媒体文件的方法，其中所述目标控制数据采取消息包围在目标控制包中的形式并且代表下列的参数：用于修饰视频和图形目标，用于定义所述各个目标的交互行为，用于产生到所述目标和从所述目标的超链接，用于定义所述目标的动画路径，用于定义动态媒体组合参数，用于分配各个值到用户变量，用于重新定向或重新定标与各个目标交互和从一个目标到另一个目标的其他控制的结果，用于附加可执行的行为到各个目标，包括语音呼叫和启动与停止定时器，以及用于定义控制动作执行的各个条件。

9. 按照权利要求 7 的产生交互多媒体文件的方法，其中修饰参数代表目标的透明度、比例、音量、位置、Z-序、背景色和旋转，在此，所述动画路径影响任何所述修饰参数，所述超链路支持到另外视频文件、在一个文件中的单个场景、和在一个场景中的作为对象的其他目标流的非线性视频和链路，所述交互行为数据包括播放的暂停和循环播放、返回用户信息到服务器、激活或者去激活目标动画、定义菜单、和可以寄存各个用户选择的简单形式。

10. 按照权利要求 7 的产生交互多媒体文件的方法，其中提供修饰动作或目标行为的有条件执行，各个条件采取定时器事件、用户事件、系统事件、交互事件、各个目标之间关系事件、用户变量、和诸如播放、暂停、发码流或独立播放之类的各个系统状态。

11. 一种从非静止三维数据集到一维的实时映射方法，包括以下各个步骤：

预计算所述数据，编码所述映射；

发送编码的映射到客户机；和

所述客户机应用所述映射到所述数据。

12. 按照权利要求 11 的从非静止三维数据集到一维的实时映射方法，其中所述数据集包括色视频帧和所述预计算包括矢量量化处理；在映射处理中对于每个单元确定最接近的码簿矢量；



利用 8 元树表示执行所述编码；  
发送所述编码的 8 元树到解码器； 和  
然后所述解码器对所述数据集施加映射。

13. 一种交互多媒体文件格式，该格式含有单个的包含视频、文本、音频、音乐、和/或图形数据的单个目标，其中所述各个目标的至少一个包括数据流，并且所述数据流的至少一个包括一个场景，所述场景的至少一个包括文件，并且其中目录数据和元数据提供文件信息。

14. 一种在面向目标交互视频系统中动态改变显示视频的实际内容的系统，包括：

动态媒体组合处理，包括包含各个目标的交互多媒体文件格式，该目标含有视频、文本、音频、音乐、和/或图形数据，其中所述各个目标的至少一个包含数据流，所述数据流的至少一个包含场景，所述场景的至少一个包含文件；

用于提供文件信息的目录数据结构；

用于允许要组合在一起的各个目标正确组合的选择机制；

用于根据所述目录信息利用目录信息并了解所述目标位置的数据流管理器；

用于在由用户收视的同时，实时地插入、删除、或替换所述场景中的所述目标和所述视频中的场景的控制机制；

15. 按照权利要求 14 的系统，包括非连续访问能力的远端服务器，用于从每个目标流中选择适当数据部分的选择机制，用于放置所述数据部分到最后组合的数据流中的交叉机制，和用于发送所述最后组合的流到客户机的无线传输机制。

16. 按照权利要求 14 的系统，包括非连续访问能力的远端服务器，包括用于执行从所述远端服务器传送到所述系统的库管理指令的机制，所述服务器能够询问所述库和接收关于其中含有规定目标的信息，并且插入、更新、或删除所述库内容；并且如果要求的话，所述动态媒体组合引擎能够同时从所述库和远端服务器两者来源目标数据流。

17. 按照权利要求 14 的系统，包括提供离线播放模式的本地服务

器：

用于在本地文件中存储适当数据部分的存储机制；

用于从分别的源选择适当数据部分的选择机制；

本地数据文件包括用于连续存储在所述文件中的每个场景的多个流；

用于所述本地服务器随机地访问所述场景中的每个流的访问机制；

用于选择进行修饰的所述目标的选择机制；

用于能够从所述远端服务器管理的动态媒体组合的持久目标库，利用全数字权限管理信息所述目标能够存储在所述库中；

可用于客户机的软件，用于执行从远端服务器传送库管理指令到客户机，所述服务器能够询问所述库并接收关于包含在其中的特定目标的信息，并且插入、更新、或删除所述库的内容；和

所述动态媒体组合引擎能够从所述库和远端服务器两者同时来源目标数据流。

18. 按照权利要求 14 的系统，其中每个所述流包括一个流结束包，用于划界流边界，在所述场景中的所述第一流含有所述场景中的所述目标的描述；

所述场景中的目标控制包提供用于与不同流的特定目标交互、改变源数据的信息；

在所述服务器中的读出机制，用于当执行本地重放时，从所述文件中同时读出多于一个的流；和

用于管理阵列或各流的链接表的机制，数据流管理器能够从每个流中按循环方式读出一个包；用于在所述文件中存储当前位置的存储机制；和用于存储各个参考目标的表的存储机制。

19. 按照权利要求 14 的系统，其中数据被码流发送到媒体播放机客户机，所述客户机能够解码从远端服务器接收的包和送回用户操作到所述服务器，所述服务器响应于用户的诸如点击之类的操作，并且修改发送到客户机的所述数据，每个所述场景含有由一个或多个目标

组成的单一的被复用的流，所述服务器根据客户机请求通过复用多个目标数据流能够实时地组合各个场景，以对任何给度场景构成一个单一复用的流，并且无线发送码流到所述客户机用于重放。

20. 按照权利要求 14 的系统，包括用于同时播放多个视频目标的播放机制，每个所述视频目标能够不同源始发，所述服务器能够打开每个所述源、交叉比特流、增加适当控制信息和传送新的组合流到所述客户机。

21. 按照权利要求 14 的系统，包括数据源管理器，能够随机访问所述源文件、从需要组合显示场景的所述各个流中读出正确数据和控制包，并且包括服务器复用器，能够从具有单一输入端的多源管理器和从所述动态媒体组合引擎接收输入，所述复用器能够从所述源复用各个目标数据在一起并插入附加的控制包到所述数据流中，用于控制组合的场景中的各个组成目标的修饰。

22. 按照权利要求 14 的系统，包括 XML 分析器，能够通过 IAVML 源程序，可编程地对所述动态媒体组合进行控制。

23. 按照权利要求 14 的系统，其中所述远端服务器能够从服务器操作员接受多个输入，以进一步控制和订制所述动态媒体组合处理，所述输入包括用户概况、人口统计、地理位置、或一天的时间。

24. 按照权利要求 14 的系统，其中所述远端服务器能够从服务器操作员接受多个输入，以进一步控制和订制所述动态媒体组合处理，所述输入包括用户交互的记录，诸如哪些广告对用户是成功的知识。

25. 一种面向目标交互多媒体文件，包括：

一个或多个连续的自包含场景的组合；

每个所述场景包括作为第一包的场景格式定义，和接着所述第一包的一个或多个数据流组；

每个所述数据流与含有各目标的第一数据流分开，该第一数据流可以根据由所述第一数据流中的目标控制信息规定进行动态媒体组合处理任选地解码和显示；和

每个所述数据流包括一个或多个单一自包含目标并由结束流标志

划分；每个所述目标含有其自己控制信息并且通过组合包流形成；通过编码未加工的交互多媒体数据形成的所述包流，包括：视频、文本、音频、音乐、或图形元件的至少一个或组合分别作为视频包流、文本包流、音频包流、音乐包流和图象包流。

26. 按照权利要求 25 的包含交互多媒体文件格式的面向目标视频系统，包括：

服务器软件，用于执行所述动态媒体组合处理，所述处理允许在用户收视的同时，显示的视频场景的实际内容将被实时动态地改变，并且用于插入、替换、或增加任何所述场景的任意构形的视频/音频视频目标；和

控制机制，用于由其他目标替换图象中的目标，以增加或删除图象中的目标，或形成当前场景以按照固定、自适应、或用户中介模式执行所述处理。

27. 按照权利要求 25 的面向目标的交互多媒体文件的，包括在所述场景中构成可订制地解压变换的数据。

28. 按照权利要求 25 的包含交互多媒体文件格式的面向目标视频系统，包括：

提供本地目标库支持所述处理的控制机制，所述库包括存储装置，用于存储所述处理的各个目标，控制机制能够从码流服务器管理所述库，控制机制用于对所述库目标提供版本控制，和用于使非持久库目标自动期满；和

控制机制，用于自动更新来自所述服务器的各个目标，用于对所述库目标提供多水平访问控制，并且用于对所述库目标的每个支持唯一识别、历史和状态。

29. 按照权利要求 25 的包含交互多媒体文件格式的面向目标视频系统，包括：

控制机制，用于通过立即执行所述动态媒体组合处理响应于会话中用户对所述目标的点击；和

控制机制，用于寄存用户的离线继续动作，并且用于在所述会话

结束时移动到新的超链路目的地。

30. 按照权利要求 25 的面向目标文件格式中实时发文件数据的方法，通过无线网络，因此一个场景仅包括一个码流，并且所述动态媒体组合引擎按适当的速率交叉来自其它流各个目标到所述第一流。

31. 按照权利要求 25 的面向目标文件格式中实时发文件数据的方法，通过无线网络，因此一个场景仅包括一个码流，并且所述动态媒体组合引擎按适当的速率交叉来自其它流各个目标到所述第一流。

32. 按照发活动视频内容到用户的权利要求 30 的方法，其中其它码流包括按实时编码的码流。

33. 按照发活动视频内容到用户的权利要求 31 的方法包括如下步骤：

所述用户连接到远端服务器；和

所述用户选择一个摄象机位置，收视由操作员/交换机处理的范围。

34. 按照发活动视频内容到用户的权利要求 31 的方法包括如下步骤：

所述用户连接到远端服务器；和

从全球定位系统或小区三角定位得到的所述用户的地理位置被用于自动提供摄象机位置的选择，帮助目的地的所述用户是选择。

35. 按照发活动视频内容到用户的权利要求 31 的方法包括如下步骤：

所述用户登记特殊的服务，服务提供商呼叫所述用户并自动发码流视频显示可能有潜在问题的乘车路由；

当所述用户登记时，可以选择为此目的的指定路由，并且可以帮助确定所述路由；和

所述系统跟踪所述用户的速度和位置，确定行驶方向和遵循的路由，然后所述系统可能沿着潜在路由搜索它的监视交通的摄象机表，确定是否有存在问题的区域，并且如果有任何问题存在，搜索系统通知所述用户并播放视频，以提供交通信息和状况。

36. 按照权利要求 26 的发广告方法，其中所述动态媒体组合处理根据存储在用户概况数据库中的用户自己概况信息选择各个目标。

37. 一种提供能够操作工作在码流视频系统的低功率装置的语音命令的方法，包括如下步骤：

在所述装置中捕捉用户的语音；

压缩所述语音；

插入所述压缩的语音的编码样值到用户控制包中；

发送所述压缩语音到能够处理语音命令的服务器；

所述服务器执行自动语音识别；

所述服务器映射转录的语音为命令集；

所述系统检查所述命令是由所述用户产生的还是由服务器产生的；

如果所述转录的命令是来自所述服务器，所述服务器执行所述命令；

如果所述转录的命令来自所述用户，所述系统传送所述命令到所述用户装置；和

所述用户执行所述命令。

38. 按照权利要求 37 的在码流视频系统中能够低功率工作的提供语音命令的方法，其中；

所述系统确定转录的命令是否是预定的；

如果所述转录的命令不是预定的，所述系统发送所述转录的文本串到所述用户；和

所述用户插入所述文本串到适当的文本字段。

39. 一种图象处理方法，包括如下步骤：

根据图象的色产生色图；

利用色图确定图象的表示；

确定利用色图表示的图象的至少一部分的相对运动。

40. 按照权利要求 39 的方法，还包括编码图象的表示的步骤。

41. 按照权利要求 39 的方法，还包括编码相对运动的步骤。

42. 按照权利要求 39 的方法, 还包括编码图象的表示和相对运动的步骤。

43. 按照权利要求 39 的方法, 其中所述产生步骤包括执行色量化, 以便产生色图。

44. 按照权利要求 43 的方法, 其中所述产生步骤还包括根据以前确定的最近帧的色图产生色图。

45. 按照权利要求 44 的方法, 其中所述产生步骤包括根据以前确定的色图重组色图, 以便使转移到当前帧的来自最近帧的像素的色被映射到色图的相同索引。

46. 按照权利要求 44 的方法, 其中所述产生步骤包括将该色图与以前确定的色图进行相关。

47. 按照权利要求 39 的方法, 其中所述确定相对运动的步骤包括确定图象的至少一部分的运动矢量。

48. 一种图象处理方法, 包括产生 4 元树, 用于编码图象的表示。

49. 按照权利要求 48 的方法, 其中编码步骤包括产生具有透明叶表示的 4 元树。

50. 按照权利要求 49 的方法, 其中编码步骤包括产生 4 元树, 具有用于代表任意构形目标的透明叶表示。

51. 按照权利要求 50 的方法, 其中编码步骤包括产生具有底部级别节点类型消除的 4 元树。

52. 一种确定图象的编码表示方法, 包括:

分析用于代表色的多个比特;

当用于代表色的比特数超过第一值时, 利用第一标志值和第一预定比特数代表色。

当用于代表色的比特数不超过第一值时, 利用第二标志值和第二预定比特数代表色;

53. 按照权利要求 52 的方法, 其中利用第一标志值代表色的步骤包括利用是 8 的第一预定数代表色; 和

利用第二标志值代表色的步骤包括利用是 4 的第二预定比特数代

表色。

54. 一种图象处理系统，包括根据图象的色产生色图的装置；  
用于利用色图确定图象的表示的装置；

用于确定利用色图表示的图象的至少一部分的相对运动的装置。

55. 按照权利要求 54 的系统，还包括用于编码图象的表示的装置。

56. 按照权利要求 54 的系统，还包括用于编码相对运动的装置。

57. 按照权利要求 54 的系统，还包括编码图象的表示和相对运动的装置。

58. 按照权利要求 54 的系统，其中所述产生装置包括执行色量化的装置，以便产生色图。

59. 按照权利要求 58 的装置，其中所述产生装置还包括根据以前确定的最近帧的色图产生色图的装置。

60. 按照权利要求 59 的装置，其中所述产生装置包括根据以前确定的色图重组色图的装置，以便使转移到当前帧的来自最近帧的像素的色被映射到色图的相同索引。

61. 按照权利要求 59 的方法，其中所述产生装置包括将该色图与以前确定的色图进行相关的装置。

62. 按照权利要求 54 的方法，其中所述确定相对运动的装置包括确定图象的至少一部分的运动矢量的装置。

63. 一种图象编码系统，包括产生 4 元树的装置，用于编码图象的表示。

64. 按照权利要求 63 的方法，其中编码装置包括产生具有透明叶表示的 4 元树的装置。

65. 按照权利要求 64 的方法，其中编码装置包括产生 4 元树的装置，具有用于代表任意构形目标的透明叶表示。

66. 按照权利要求 65 的方法，其中编码装置包括产生具有底部级别节点类型消除的 4 元树的装置。

67. 一种确定图象的编码表示的图象编码系统，包括：  
分析用于代表色的多个比特的装置；



当用于代表色的比特数超过第一值时，利用第一标志值和第一预定比特数代表色的装置；

当用于代表色的比特数不超过第一值时，利用第二标志值和第二预定比特数代表色的装置。

68. 按照权利要求 67 的系统，其中利用第一标志值代表色的步骤包括利用是 8 的第一预定数代表色的装置；和

利用第二标志值代表色的步骤包括利用是 4 的第二预定比特数代表色的装置。

69. 一种处理目标的方法，包括以下步骤：

分析源程序语言中的信息；

读出多个含有多个目标的数据源，这些目标以视频、图形、动画、和音频至少之一的形式；

根据源程序语言中的信息附加控制信息到多个目标上；

交叉多个目标到数据流和文件的至少一个。

70. 按照权利要求 69 的方法，还包括从用户输入信息的步骤，其中根据源程序语言中的信息和来自用户的信息执行该附加的步骤。

71. 按照权利要求 69 的方法，还包括输入从概况信息、人口统计信息、地理信息、和暂时信息的至少一个中选择的控制信息的步骤，其中根据源程序语言中的信息和控制信息执行该附加的步骤。

72. 按照权利要求 71 的方法，还包括从用户输入信息的步骤，其中根据源程序语言中的信息、控制信息、和来自用户的信息执行该附加的步骤。

73. 按照权利要求 72 的方法，其中从用户输入信息的步骤包括在显示器上图形指示和选择一个目标。

74. 按照权利要求 69 的方法，还包括插入目标到数据流和文件的至少一个中的步骤。

75. 按照权利要求 74 的方法，其中所述插入步骤包括插入广告到数据流和文件的至少一个。

76. 按照权利要求 75 的方法，还包括利用不同目标替代广告的步骤

骤。

77. 按照权利要求 74 的方法，其中所述插入步骤包括插入图形字符到数据流和文件的至少一个。

78. 按照权利要求 77 的方法，其中所述插入图形字符的步骤包括根据用户的地理位置插入图形字符。

79. 按照权利要求 69 的方法，还包括利用不同目标替代多个目标之一的步骤。

80. 按照权利要求 79 的方法，其中所述替换多个目标之一的步骤包括利用新的场景替换收视场景的多个目标之一的步骤。

81. 按照权利要求 69 的方法，其中所述读出多个数据源的步骤包括读出多个训练视频的至少一个的数据源。

82. 按照权利要求 69 的方法，其中所述读出多个数据源的步骤包括读出多个教育视频的至少一个的数据源。

83. 按照权利要求 69 的方法，其中所述读出多个数据源的步骤包括读出多个推销宣传视频的至少一个的数据源。

84. 按照权利要求 69 的方法，其中所述读出多个数据源的步骤包括读出多个娱乐视频的至少一个的数据源。

85. 按照权利要求 69 的方法，其中所述读出多个数据源的步骤包括从监视摄象机获得视频。

86. 按照权利要求 74 的方法，其中所述插入步骤包括插入来自摄象机的视频，用于将监视的车辆交通数据加入到数据流和文件至少之一中。

87. 按照权利要求 74 的方法，其中所述插入步骤包括插入贺卡信息到数据流和文件至少之一中。

88. 按照权利要求 74 的方法，其中所述插入步骤包括插入远端计算装置的监视器的计算机产生的图象。

89. 按照权利要求 69 的方法，还包括提供数据流和文件至少之一到用户的步骤，其中数据流和文件至少之一包括交互视频节目集。

90. 按照权利要求 69 的方法，还包括提供包含交互形式的数据流

和文件至少之一到用户的步骤;

由用户电子地填写的表格; 和

当填写表格时, 电子地存储由用户输入的信息。

91. 按照权利要求 90 的方法, 还包括传输已被电子存储的信息的步骤。

92. 按照权利要求 69 的方法, 其中附加控制信息的步骤包括附加指示交互行为的控制信息。

93. 按照权利要求 69 的方法, 其中附加控制信息的步骤包括附加含修饰参数的控制信息。

94. 按照权利要求 69 的方法, 其中附加控制信息的步骤包括附加含组合信息的控制信息。

95. 按照权利要求 69 的方法, 其中附加控制信息的步骤包括附加指示如何处理压缩数据的控制信息。

96. 按照权利要求 69 的方法, 其中附加控制信息的步骤包括附加可执行的行为。

97. 按照权利要求 96 的方法, 其中附加可执行的行为包括附加用于动画的修饰参数。

98. 按照权利要求 96 的方法, 其中附加可执行的行为包括附加超链路。

99. 按照权利要求 96 的方法, 其中附加可执行的行为包括附加定时器。

100. 按照权利要求 96 的方法, 其中附加可执行的行为包括附加允许进行语音呼叫的行为。

101. 按照权利要求 96 的方法, 其中附加可执行的行为包括附加包含至少暂停和播放之一的系统状态。

102. 按照权利要求 96 的方法, 其中附加可执行的行为包括附加允许改变用户变量的信息。

103. 一种处理目标的系统, 包括:

用于分析按源程序语言信息的装置;

用于读出含有多个以视频、图形、动画、和音频至少一个的形式的目标的多个数据源的装置；

用于根据源程序语言的信息附加控制信息到多个目标上的装置；

用于交叉多个目标到数据流和文件至少之一的装置。

104. 按照权利要求 103 的系统，还包括用于从用户输入信息的装置，其中用于附加的装置根据源程序语言的信息和控制信息操作。

105. 按照权利要求 103 的系统，还包括输入从概况信息、人口统计信息、地理信息、和暂时信息的至少之一选择的控制信息的装置，其中用于附加的装置根据源程序语言的信息和控制信息操作。

106. 按照权利要求 105 的系统，还包括用于输入来自用户信息的装置，其中用于附加的装置根据源程序语言的信息、控制信息、和来自用户的信息操作。

107. 按照权利要求 106 的系统，其中输入用户信息的装置包括在显示器上图形指示和选择目标的装置。

108. 按照权利要求 103 的系统，还包括用于插入目标到数据流和文件之一的装置。

109. 按照权利要求 108 的系统，其中所述插入装置包括插入广告到数据流和文件的至少一个的装置。

110. 按照权利要求 109 的系统，还包括利用不同目标替代广告装置。

111. 按照权利要求 108 的系统，其中所述插入装置包括插入图形字符到数据流和文件的至少一个的装置。

112. 按照权利要求 111 的系统，其中所述插入图形字符的装置包括根据用户的地理位置插入图形字符的装置。

113. 按照权利要求 103 的系统，还包括利用不同目标替代多个目标之一的装置。

114. 按照权利要求 113 的系统，其中所述替换多个目标之一的装置包括利用新的场景替换收视场景的多个目标之一的装置。

115. 按照权利要求 103 的系统，其中所述读出多个数据源的装置

包括读出多个训练视频的至少一个的数据源的装置。

116. 按照权利要求 103 的系统，其中所述读出多个数据源的装置包括读出多个推销宣传视频的至少一个的数据源的装置。

117. 按照权利要求 103 的系统，其中所述读出多个数据源的装置包括读出多个娱乐视频的至少一个的数据源的装置。

118. 按照权利要求 103 的系统，其中所述读出多个数据源的装置包括读出多个教育视频的至少一个的数据源的装置。

119. 按照权利要求 103 的系统，其中所述读出多个数据源的装置包括从监视摄像机获得视频的装置。

120. 按照权利要求 107 的系统，其中所述插入装置包括插入来自摄像机的视频，用于将监视的车辆交通数据加入到数据流和文件至少之一中的装置。

121. 按照权利要求 107 的系统，其中所述插入装置包括插入贺卡信息到数据流和文件至少之一中的装置。

122. 按照权利要求 107 的系统，其中所述插入装置包括插入远端计算装置的监视器的计算机产生的图象。

123. 按照权利要求 103 的系统，还包括提供数据流和文件至少之一到用户的装置，其中数据流和文件至少之一包括交互视频节目集。

124. 按照权利要求 103 的系统，还包括提供包含交互形式的数据流和文件至少之一到用户的装置；

用于由用户电子地填写表格的装置；和

当填写表格时，电子地存储由用户输入的信息的装置。

125. 按照权利要求 124 的系统，还包括用于传输已被电子存储的信息的装置。

126. 按照权利要求 103 的系统，其中附加控制信息的装置包括附加指示交互行为的控制信息的装置。

127. 按照权利要求 103 的系统，其中附加控制信息的装置包括附加含修饰参数的控制信息的装置。

128. 按照权利要求 103 的系统，其中附加控制信息的装置包括附

加含组合信息的控制信息的装置。

129. 按照权利要求 103 的系统，其中附加控制信息的装置包括附加指示如何处理压缩数据的控制信息的装置。

130. 按照权利要求 103 的系统，其中附加控制信息的装置包括附加可执行的行为的装置。

131. 按照权利要求 130 的系统，其中附加可执行的行为的装置包括附加用于动画的修饰参数的装置。

132. 按照权利要求 130 的系统，其中附加可执行的行为的装置包括附加超链路的装置。

133. 按照权利要求 130 的系统，其中附加可执行的行为的装置包括附加定时器的装置。

134. 按照权利要求 130 的系统，其中附加可执行的行为的装置包括附加允许进行语音呼叫的行为的装置。

135. 按照权利要求 130 的系统，其中附加可执行的行为的装置包括附加包含至少暂停和播放之一的系统状态的装置。

136. 按照权利要求 130 的系统，其中附加可执行的行为的装置包括附加允许改变用户变量的信息的装置。

137. 一种远端控制计算机的方法，包括如下步骤：

在服务器根据数据执行计算操作；

根据计算操作在服务器产生图象信息；

经无线连接，从服务器发送图象信息到客户机计算装置，而不发送所述数据；

由客户机计算装置接收图象信息；和

由客户机计算装置显示图象信息。

138. 按照权利要求 137 的方法，还包括由客户机计算装置的用户输入输入信息的步骤；

经无线连接，发送从客户机计算装置输入的信息到服务器；

在服务器处理输入信息；

在服务器根据输入信息改变图象信息；

经无线连接，发送已经被改变的图象信息；  
由客户机计算装置接收已经被改变的图象信息；和  
由客户机计算装置显示已经被改变的图象信息。

139. 按照权利要求 137 的方法，还包括在服务器捕捉图象信息的步骤，其中发送步骤包括发送已经被捕捉的图象信息。

140. 按照权利要求 137 的方法，其中发送步骤包括发送作为具有附加上控制信息的视频目标的图象信息。

141. 一种远端控制计算机的系统，包括  
在服务器根据数据执行计算操作的装置；  
根据计算操作在服务器产生图象信息的装置；  
经无线连接，从服务器发送图象信息到客户机计算装置，而不发送所述数据的装置；

由客户机计算装置接收图象信息的装置；和  
由客户机计算装置显示图象信息的装置。

142. 按照权利要求 141 的系统，还包括由客户机计算装置的用户输入输入信息的装置；

经无线连接，发送从客户机计算装置输入的信息到服务器的装置；  
在服务器处理输入信息的装置；  
在服务器根据输入信息改变图象信息的装置；  
经无线连接，发送已经被改变的图象信息的装置；  
由客户机计算装置接收已经被改变的图象信息的装置；和  
由客户机计算装置显示已经被改变的图象信息的装置。

143. 按照权利要求 141 的系统，还包括在服务器捕捉图象信息的装置，其中发送装置包括：

发送已经被捕捉的图象信息的装置。

144. 按照权利要求 139 的系统，其中发送装置包括发送作为具有附加上控制信息的视频目标的图象信息的装置。

145. 一种发送电子贺卡的方法，包括如下步骤：

输入指示贺卡特征的信息；

产生对应于贺卡的图象信息；  
编码该图象信息，作为具有控制信息的目标；  
通过无线连接发送具有控制信息的目标；  
由无线手持计算装置接收具有控制信息的目标；  
由无线手持计算装置解码具有控制信息的目标为贺卡图象；和  
在无线手持计算装置上显示已经被解码的贺卡图象。

146. 按照权利要求 145 的方法，其中产生图象信息的步骤包括捕捉至少一个图象和作为订制图象信息的一系列图象，其中编码步骤还包括编码所述订制图象作为具有控制信息的目标，其中所述解码步骤包括解码利用图象信息编码的目标和解码利用订制图象信息编码的目标，其中所述显示步骤包括显示图象信息和订制图象信息作为贺卡。

147. 一种发送电子贺卡的系统，包括：

用于输入指示贺卡特征信息的装置；  
用于产生对应于该贺卡的图象信息的装置；  
用于编码图象信息为一个具有控制信息的目标的装置；  
用于通过无线手持计算装置接收具有控制信息的目的装置；  
用于通过无线连接传输该具有控制信息的目标的装置；  
用于通过无线手持计算装置解码具有控制信息的目标为贺卡图象的装置；和  
用于显示在手持计算装置中已经解码的贺卡图象。

148. 按照权利要求 147 的方法，其中产生图象信息的装置包括捕捉至少一个图象和作为订制图象信息的一系列图象的装置，其中编码装置还包括编码所述订制图象作为具有控制信息的目标的装置，其中所述解码装置包括解码利用图象信息编码的目标和解码利用订制图象信息编码的目标的装置，其中所述显示装置包括显示图象信息和订制图象信息作为贺卡的装置。

149. 一种控制计算装置的方法，包括如下步骤：

通过计算装置输入音频信号；  
编码音频信号；



发送音频信号到远端计算装置;

在远端计算装置翻译音频信号和产生对应于该音频信号的信息;

发送对应于该音频信号的信息到计算装置;

利用对应于该音频信号的信息控制计算装置。

150. 按照权利要求 149 的方法, 其中所述控制步骤包括利用对应于信息的计算机指令控制计算装置, 该信息对应于该音频信号。

151. 按照权利要求 149 的方法, 其中所述控制步骤包括利用对应于该信息的数据控制计算装置, 该信息对应于该音频信号。

152. 按照权利要求 149 的方法, 其中翻译音频信号的步骤包括执行语音识别。

153. 一种控制计算装置的系统, 包括:

通过计算装置输入音频信号;

编码音频信号;

发送音频信号到远端计算装置;

在远端计算装置翻译音频信号和产生对应于该音频信号的信息;

发送对应于该音频信号的信息到计算装置; 和

利用对应于该音频信号的信息控制计算装置。

154. 按照权利要求 153 的系统, 其中所述控制步骤包括利用对应于信息的计算机指令控制计算装置的装置, 该信息对应于该音频信号。

155. 按照权利要求 153 的系统, 其中所述控制装置包括利用对应于该信息的数据控制计算装置的装置, 该信息对应于该音频信号。

156. 按照权利要求 153 的方法, 其中翻译音频信号的装置包括执行语音识别的装置。

157. 一种执行发送的方法, 包括如下步骤:

在无线手持装置上显示广告;

从无线手持装置发送信息; 和

接收与由于显示广告已经发送的信息相关的打折价格。

158. 按照权利要求 157 的方法, 其中显示步骤是在发送步骤之前执行的。

159. 按照权利要求 157 的方法，其中显示步骤是在发送步骤期间执行的。

160. 按照权利要求 157 的方法，其中显示步骤是在发送步骤之后执行的。

161. 按照权利要求 157 的方法，其中接收打折价格的步骤包括与已经发送的信息相关的整个费用的打折。

162. 按照权利要求 157 的方法，其中显示步骤包括显示作为交互目标的目标，该方法还包括由用户与该目标交互，并响应于由用户进行的交互显示视频。

163. 一种执行发送的系统，包括

在无线手持装置上显示广告的装置；

从无线手持装置发送信息的装置；和

接收与由于显示广告已经发送的信息相关的打折费用的装置。

164. 按照权利要求 163 的系统，其中显示广告的装置是在信息发送之前操作的。

165. 按照权利要求 163 的系统，其中显示广告的装置是在信息发送期间操作的。

166. 按照权利要求 163 的系统，其中显示广告的装置是在信息发送之后操作的。

167. 按照权利要求 163 的系统，其中接收打折价格的装置包括接收与已经发送的信息相关的整个费用的打折的装置。

168. 按照权利要求 163 的系统，其中显示装置包括显示作为交互目标的目标的装置，该系统还包括由用户与该目标交互的装置，和响应于由用户进行的交互显示视频的装置。

169. 一种提供视频的方法，包括如下步骤：

确定是否发生了事件；

响应于该事件，获得通过无线发送该区域的视频到用户的一个区域的视频。

170. 按照权利要求 169 的方法，其中确定的步骤包括由用户选择

一个位置。其中发送的步骤包括发送对应于所述位置的区域的视频。

171. 按照权利要求 170 的方法，其中选择的步骤包括拨号对应于交通视频的电话号码。

172. 按照权利要求 169 的方法，还包括执行利用全球定位系统确定区域的步骤。

173. 按照权利要求 169 的方法，还包括根据由用户使用的小区场地执行确定区域的步骤。

174. 按照权利要求 169 的方法，其中确定步骤包括确定在预定路由存在交通问题，其中获得视频步骤包括获得对应于该预定路由的视频。

175. 按照权利要求 174 的方法，其中发送步骤包括仅当用户的移动速度大于预定速度时，发送视频到该用户。

176. 一种提供视频的系统，包括：

确定是否发生了事件的装置；

获得一个区域的视频的装置；

响应于该事件，通过无线发送该区域的视频到用户的装置。

177. 按照权利要求 176 的系统，其中确定的装置包括由用户选择一个位置的装置。其中发送的装置包括发送对应于所述位置的区域的视频的装置。

178. 按照权利要求 177 的系统，其中选择的装置包括拨号对应于交通视频的电话号码的装置。

179. 按照权利要求 176 的系统，还包括执行利用全球定位系统确定区域的装置。

180. 按照权利要求 176 的系统，还包括根据由用户使用的小区场地执行确定区域的装置。

181. 按照权利要求 176 的系统，其中确定装置包括确定在预定路由存在交通问题的装置，其中获得视频装置包括获得对应于该预定路由的视频的装置。

182. 按照权利要求 181 的系统，其中发送装置包括仅当用户的移

动速度大于预定速度时，发送视频到该用户的装置。

183. 一种面向目标多媒体视频系统，该系统能够支持多个任意构形视频目标，而不需要特别的数据额外开销或处理额外开销就能提供视频目标构形信息。

184. 按照权利要求 183 的系统，其中所述视频目标具有其自己的附加控制信息。

185. 按照权利要求 183 的系统，其中所述视频目标是从远端服务器向客户机发送的码流。

186. 按照权利要求 183 的系统，其中所述视频目标构形是按图象的表示固有地编码的。

187. 按照权利要求 69 的方法，其中附加控制信息的步骤包括附加执行控制的条件。

188. 按照权利要求 71 的方法，还包括从用户标志或变量获得信息的步骤，其中附加步骤是根据源程序语言的信息、控制信息、和来自所述用户标志的信息执行的。

189. 一种由服务器启动通信的传送多媒体内容到无线装置的方法，其中内容是针对按要求的时间或成本效率方式传送预先安排的并且经由装置的显示器或其它指示器所述用户被改变为传送完成。

190. 按照权利要求 189 的方法，其中所述用户在内容服务提供商寄存对传送特定内容的请求，所述请求被用于自动预先安排网络启动传送到客户机装置。

191. 一种交互系统，其中存储的信息可以进行离线收视和存储用户的输入，并且当所述装置下一次连接在线时交互被自动地通过无线网络传送到特定的远端服务器。

192. 按照权利要求 191 的交互系统，其中所述存储的信息是面向目标多媒体数据，该数据可以进行非线性导航。

193. 按照权利要求 69 的方法，其中所述读出多个数据源的步骤包括读出采取以下形式的多个数据源的至少一个的步骤，该各个形式是：市场、促销、产品信息、娱乐视频。

194. 按照权利要求 51 的方法，其中编码步骤包括产生具有叶节点值的 4 元树，如果标志被定义为真，则表示为 FIFO 缓冲器的索引，或者如果标志被定义为假，则表示为色值。

195. 按照权利要求 66 的系统，其中编码的装置包括用于产生具有叶节点值的 4 元树的装置，如果标志被定义为真，则表示为 FIFO 缓冲器的索引，或者如果标志被定义为假，则表示为色值。

196. 按照权利要求 51 的方法，其中编码步骤包括产生具有叶节点值的 4 元树，表示为平均值加水平和垂直梯度。

197. 按照权利要求 196 的方法，其中编码步骤包括产生具有叶节点值的 4 元树，如果标志被定义为真，则表示为 FIFO 缓冲器的索引，或者如果标志被定义为假，则表示为色值。

198. 按照权利要求 66 的系统，其中编码装置包括产生具有叶节点值的 4 元树，表示为平均值加水平和垂直梯度。

199. 按照权利要求 198 的系统，其中编码的装置包括用于产生具有叶节点值的 4 元树的装置，如果标志被定义为真，则表示为 FIFO 缓冲器的索引，或者如果标志被定义为假，则表示为色值。

200. 按照权利要求 14 的系统，包括在便携客户机装置上用于动态媒体组合中使用的持久目标库，所述库能从所述远端服务器进行管理，可用于客户机的用于执行可以从所述远端服务器传送到客户机的库管理指令的软件，所述服务器能够询问所述库和接收包含在其中的关于特定目标的信息，并且可以插入、更新、或删除所述库的内容，并且所述动态媒体组合引擎能够同时从所述库和远端服务器两者来源目标数据流。如果需要，所述持久目标库存储目标信息，包括：期满日期、访问许可、唯一识别符、元数据和状态信息，所述系统执行对过期目标的自动垃圾收集、访问控制、库检索、和各种其它库管理任务。

201. 一种视频编码方法，包括：

编码具有目标控制数据的视频数据作为视频目标；和

产生包含多个所述视频目标的数据流，该视频目标具有相应的视

频数据和目标控制数据。

202. 按照权利要求 201 的视频编码方法，包括：

产生表示场景的场景包和包括具有相应各个视频目标的多个所述数据流。

203. 按照权利要求 202 的视频编码方法，包括产生包含多个所述场景包的视频数据文件，该场景包具有相应数据流和用户控制数据。

204. 按照权利要求 201 的视频编码方法，其中所述视频数据代表视频帧、音频帧、文本和/或图形。

205. 按照权利要求 201 的视频编码方法，其中所述视频目标包括一个包，该包具有所述编码视频数据的各个数据包和具有用于所述视频目标的所述目标控制数据的至少一个目标控制包。

206. 按照权利要求 202 的视频编码方法，其中所述视频数据文件、所述场景包、和所述数据流包括相应目录数据。

207. 按照权利要求 201 的视频编码方法，其中所述控制数据代表定义所述视频目标的各个参数，允许由用户在一个场景内交互控制所述目标。

208. 按照权利要求 201 的视频编码方法，其中所述编码包括利用代表所述视频目标的构形的构形数据编码所述视频数据的亮度和色信息。

209. 按照权利要求 201 的视频编码方法，其中所述目标控制数据对所述各个视频目标定义构形、修饰、动画和交互参数。

210. 一种视频编码方法，包括：

根据减少的色表示量化视频流中的色数据；

产生代表所述量化的色和透明区的编码的视频帧数据；和

产生编码的音频数据和目标控制数据，用于随着所述编码的视频数据传输。

211. 按照权利要求 210 的视频编码方法，包括：

产生代表所述流的视频帧中的色变化的运动矢量；所述编码的视频帧数据代表所述运动矢量。

212. 按照权利要求 211 的视频编码方法，包括：

产生编码的文本目标和矢量图形目标和音乐目标数据，用于随着所述编码的视频数据传输；和

产生编码的数据，用于配置可订制地解码变换。

213. 按照权利要求 2 的视频编码方法，包括根据用户与所述视频目标的交互，为用户动态地实时产生所述场景包。

214. 按照权利要求 1 的视频编码方法，其中所述目标控制数据代表如下各个参数：(i) 修饰视频目标，(ii) 定义所述目标的交互行为，(iii) 产生到和从所述目标的超链路，(iv) 为所述目标定义动画路径，(v) 定义动态媒体组合参数，(vi) 分配值给用户变量，和/或 (vii) 定义执行控制动作的条件。

215. 按照权利要求 210 或 211 的视频编码方法，其中所述目标控制数据代表用于修饰视频帧的目标的参数。

216. 按照权利要求 210 或 211 的视频编码方法，其中所述参数表示透明、比例、音量、位置、和旋转。

217. 按照权利要求 210 或 211 的视频编码方法，其中所述编码的视频、音频、和控制数据是作为用于各自解码的相应包发送的。

218. 一种视频编码方法，包括：

(i) 选择视频数据的每个视频帧的减少的色集；

(ii) 逐帧地调和色；

(iii) 执行运动补偿；

(iv) 根据感觉色差测量确定帧的更新区；

(v) 根据步骤 (i) 到 (iv) 编码所述帧的视频数据为视频目标；

和

(vi) 在每个视频目标中包括动画、修饰、和动态组合控制。

219. 一种视频解码方法，用于解码按照在先的权利要求的任何一个所要求的方法编码的视频数据。

220. 按照权利要求 219 的视频解码方法，包括分析所述编码的数据，分配目标控制包到目标管理处理和分配编码的视频包到视频解码

器。

221. 按照权利要求 214 的视频编码方法, 其中所述修饰参数表示目标的透明度、比例、音量、位置和旋转。

222. 按照权利要求 214 的视频编码方法, 其中所述动画路径调节所述修饰参数。

223. 按照权利要求 214 的视频编码方法, 其中所述超链路表示到相应视频文件、场景包和目标的链路。

224. 按照权利要求 214 的视频编码方法, 其中所述交互行为数据提供对所述目标的播放、和用户数据返回的控制。

225. 按照权利要求 220 的视频解码方法, 包括根据所述目标控制包产生对用户的视频目标控制, 用于接收和修饰视频目标。

226. 一种用于执行按照权利要求 219 所要求的视频解码方法的视频解码器。

227. 一种具有按照权利要求 226 所要求的视频解码器的计算机装置。

228. 一种按照权利要求 227 所要求的计算机装置, 其中所述装置是便携的和手持的, 例如移动电话或 PDA。

229. 一种动态色空间编码方法, 包括执行按照权利要求 1 色要求的视频编码方法和发送到用户的添加的附加色量化信息, 能使所述用户选择实时色减少。

230. 按照权利要求 201 的视频编码方法, 包括添加定标用户和/或具有所述视频目标的本地视频广告。

231. 一种具有超薄客户机的计算机装置, 用于执行按照权利要求 219 所要求的视频解码方法并且自适应地访问包括所述视频目标的远端服务器。

232. 一种多方电视会议的方法, 包括执行按照权利要求 201 所要求的视频编码方法。

233. 按照权利要求 201 的视频编码方法, 包括产生用于用户选择所述视频目标的视频菜单和表格。



234. 一种产生传输到移动电话的电子卡的方法，包括执行按照权利要求 201 所要求的所述视频编码方法。

235. 一种具有用于执行按照权利要求 201 到 218 的任何一个视频编码方法的步骤的视频编码器。

236. 一种视频点播系统，包括按照权利要求 235 所要求的视频编码器。

237. 一种安全系统，包括按照权利要求 235 所要求的视频编码器。

238. 一种交互移动视频系统，包括按照权利要求 226 所要求的视频解码器。

239. 一种按照权利要求 219 所要求的视频解码方法，包括处理来自用户的语音命令，以控制基于所述视频目标产生的视频显示。

240. 一种存储在计算机可读存储媒体中的计算机程序，包括用于执行按照权利要求 219 所要求的视频解码方法的代码并且产生包含所述视频目标的控制的视频显示，并且响应于所述控制的应用调节所述显示。

241. 一种按照权利要求 240 所要求的计算机程序，包括 IAVML 指令。

242. 一种无线发送视频码流和动画系统，包括：

(i) 便携监视装置和第一无线通信装置；

(ii) 服务器，用于存储压缩的数字视频、和计算机动画，并且能使用户从可用视频库中浏览和选择收视的数字视频；和

(iii) 至少一个接口模块，包含第二无线通信装置，用于从服务器发送可发送的数据到便携监视装置，该便携监视装置包括用于接收所述可发送数据的装置，变换可发送数据为显示视频图象的视频图象，并且允许用户与服务器通信，进行交互地浏览和选择一个收视的视频。

243. 按照权利要求 242 的无线发送视频码流和动画系统，其中所述便携无线装置是手持处理装置。

244. 一种提供无线发送视频码流和动画的方法，至少包括如下各步骤之一：

(a)通过广域网下载并存储来自远端服务器的压缩视频和动画数据，用于以后从本地服务器发送；

(b)允许用户浏览和选择来自存储在本地服务器的视频数据库的数字视频数据进行收视；

(c)发送数据到便携监视装置；和

(d)在便携监视装置上处理该数据，以显示图象。

245. 一种提供交互视频节目集的方法，至少包括如下各步骤之一：

(a)通过规定如下产生视频节目集：(i)在节目集中的各种场景和在每个场景可能出现的各种视频目标，(ii)规定预置和用户可选场景导航控制和对每个场景的单个组合规则，(iii)对各个媒体目标规定修饰参数，(iv)规定对各个媒体目标的控制，产生收集用户反馈的表格，(v)集成压缩的媒体流和目标控制信息为组合数据流。

246. 按照权利要求 245 的方法，包括：

(a)处理组合的数据流并翻译目标控制信息以显示每个场景；

(b)处理用户输入，以执行任何相关的目标控制，诸如通过节目集导航、激活动画等，寄存和用户选择、和其它的用户输入；

(c)存储用户选择和用户输入，用于当网络连接变为可用时加载到视频节目集网络服务器的提供商；和

(d)在远端网络服务器，从交互视频节目集接收加载的用户的选并且处理该信息以集成该信息到用户/客户机数据库。

247. 一种产生和发送视频贺卡到移动装置的方法，至少包括如下步骤之一：

(a)通过以下步骤允许用户产生视频贺卡：(i)从库组选择模板视频场景或动画，(ii)通过增加用户馈送的文本或音频模板、或从库中选择视频目标作为角色将被插入到场景中订制模板；

(b)从用户获得：(i)识别的细节，(ii)优选的传送方法，(iii)付费细节，(iv)希望接受的移动装置数；和

(c)取决于的传送方法排队贺卡，直至带宽变得可用或采取高峰

传输已过，轮询接受的装置看其是否能够处理贺卡和是否传送到指定的移动装置。

248. 按照权利要求 201 的视频编码方法，其中所述目标控制数据包括允许用户修饰对应于所述视频目标的任意构形的视频的构形参数。

249. 按照权利要求 201 的视频编码方法，其中所述目标控制数据包括确定何时调用对于所述视频目标的对应的控制的条件数据。

250. 按照权利要求 201 的视频编码方法，其中所述目标控制数据表示用于影响其它视频目标的控制。

251. 按照权利要求 201 的视频编码方法，包括响应于一些事件或用户的交互，根据设置的标志控制所述视频目标的动态媒体组合。

252. 按照权利要求 201 的视频编码方法，包括广播和/或多点传送所述数据流。

## 面向目标的视频系统

### 技术领域

本发明涉及视频编码和处理方法，更具体但不排它地涉及支持在视频场景中多个任意构形视频目标并对每个目标允许定义个别动画和交互状态，并且通过编码面向目标控制允许动态媒体成份进入可以由远端客户或独立系统解码的视频流共存的视频编码系统。客户机系统可以在标准计算机或移动计算机装置上执行，该移动计算机装置是诸如个人数字助理（PDA）、智能无线电话、手持计算机和利用低功率，通用 CPU 的耐用计算装置。这些装置可以包括对编码视频流的无线传输的支持。

### 背景技术

当前技术的进步已经导致引入了个人计算装置，这些装置刚好开始包含全面无线通信技术。无线移动电话的全面发展已经取得重大成功，但仍然具有相当大的潜在增长空间。已经认识到，尚没有对潜在新的和创新的移动视频处理提供视频质量、帧速率或低功耗的任何视频技术解决方案。由于受到移动装置的处理能力的限制，当前没有用于处理正在使用的各种个人计算装置的适合的移动视频解决办法，这些个人计算装置诸如是移动电视会议装置、超薄无线网络客户机计算装置、广播无线移动视频装置、移动视频促销或监督装置。

试图在诸如智能电话和 PDA 之类的便携手持装置上显示视频的难以解决的问题是一般这些装置具有有限的显示能力。因为视频一般利用连续色表示进行编码，要求提供真彩色（16 或 24 比特）显示能力，当使用 8 比特显示器时，导致严格执行降格操作。这是由于在客户机执行量化和抖动处理，变换视频图象为适合利用固定彩图在各装置显示的 8 比特格式，这种做法降低了质量并带来了大量的处理额外

开销。

基于电视会议的计算机当前利用标准计算机工作站或者通过网络连接的各个 PC，该网络包括物理缆线和网络计算机通信协议层。这种情况的例子是利用物理连接的端对端的缆线，并利用 TCP/IP 网络通信协议，通过互联网的两个 PC 之间的电视会议。这种类型的电视会议具有到互联网的物理连接，并且还利用大的、以计算机为基础的视频监视设备。这种系统提供各固定位置之间的电视会议，但它附加地限制了各个参加者在特定时间与会，以保证两个当事人将同时在适当位置上。

对个人手持计算机或智能电话的无线文本信息的广播，随着新的和创新的无线技术与手持计算装置的进步，近来变得切实可行。手持计算装置和移动电话能够无线连接到能提供文本信息给用户装置的广域网。目前尚没有到无线手持计算装置的视频实时传输。这种视频内容连接性的缺乏，往往限制了现存系统的商业可利用性，特别是，当一个人认为“广播”系统没有为广告的目的定标（targeted）向各个目标用户的能力时。以任何形式对广播媒体的一个重要市场发布是广告和如何得到支持的问题。有效发布广告应当是定标到用户和地理位置，但广播技术在这方面固有地受到限制。作为一种推理，支持这种系统的专业产品的“适当位置”的广告将是难以作到的。

由于考虑处理要求需要在传输期间实时地插入广告内容到视频数据流中，当前视频广播不能插入有目标的广告。在本发明人看来按照有规律的基础上将要执行的在传输之前预组合视频的替代方法太繁琐冗长。另外，一旦广告插入到视频流中，用户不能与广告交互，这样降低了广告的作用。显然，已经认识到通过交互技术可以实现更大的广告作用。

大多数编码器/解码器对于卡通或动画内容呈现差的性能，但是，对于互联网而言，正在产生比视频更多的卡通和动画内容。已经认识到，存在着能够有效地编码动画和卡通图形以及视频的编解码器的需求。

利用实现在中央位置的具有视频监视的闭路监视系统，已经实现了对数据的基于商业和家庭安全的视频监视系统，要求全时间地关注专门的监视警戒。在中央控制中心利用专用监视设备仅可以实现多位置的视频监视。安全警戒在巡检的同时不必访问来自被监视位置的视频。

利用轻型客户机工作站以网络为基础的计算包括在客户机工作站中的小型软件处理，而软件处理的多数部分发生在服务器计算机中。由于信息的集中和操作软件的配置，客户机计算降低了计算机管理的成本。客户工作站通过诸如 10 Base T 以太网之类的标准局域网进行到服务器计算机的物理连线。客户工作站运行小型操作系统，能够与后端服务器计算机和在客户机视频监视设备上的信息显示器进行通信。但是，现存的各种系统仍受到限制。它们是典型地限制了一些特殊应用或卖主的软件。例如，当前轻型客户机不能同时服务于显示视频和电子表格应用。

为了在市场上直接促销产品，销售代表可以利用视频演示说明产品和用途和性能。当前，对于移动销售商，包括使用笨重的专用视频显示设备，可以带到各个用户的地点用于产品展示。存在可得到的非移动携带式的视频显示解决方案，提供实时的视频用于产品 and 市场促销目的。

视频节目集（video brochure）经常用于市场和广告业务。但是，由于视频是一种经典的被动媒体，所以它们的效率始终受到限制。已经认识到，如果设计成交互方式，可能极大地改善视频节目集的效率。如果这种交互性可能在编解码器中固有地提供，这可能开拓以视频为基础的电子商业应用之路。对于交互视频的常规定义包括播放器，能够解压正常压缩的视频为收视窗口，并且翻译某些定义按键和叠加在视频上的隐藏“热区”的元数据，一般代表超链路，用户的鼠标点击将调用某些预定的操作。在这种典型的方法中，视频被作为来自元数据的分别的单元进行存储，并且交互的性质受到极大的限制，因为视频内容与施加的外部控制之间不存在集成。

提供交互视频的另外一种方法是允许多目标的 MPEG4, 但是, 发现这种方法在当今一般诸如具有 128Mb 的 RMA 的奔 III 500MHz 计算机之类的台式计算机上运行困难。原因是目标形状信息由目标的色度/亮度信息分别进行编码, 产生附加的存储额外开销, 并且情景描述 (BIFS) 的性质和文件格式已经部分地采取虚拟实体标记语言 (VRML) 是非常复杂的。这意味着对于视频目标的每个视频帧的 3 个分量的分量: 亮度信息、形状/透明度信息、和 BIFS 必须全解码。然后, 在目标可以被显示之前, 这些必须被混合在一起。规定以 DCT 为基础的视频编解码器本身已具有很强的计算能力, 而除了存储的额外开销以外, 附加的解码还要求引入了大量的处理额外开销。

对个人数字助理 (PDA) 的无线接入兼容性, 通过能实时地无线传送音频-视频内容到 PDA, 允许电子书籍免除了其存储方面的限制。许多团体训练的应用需要在便携装置以无线方式得到视听信息。视听训练教材的特点是交互的并且提供大量存储内容的非线性导航。利用现有技术的当前状态是不能提供的。

## 发明目的

本发明的一个目的是克服上述的各个缺点。本发明的另一个目的的提供一种播放视频流的软件并在低处理功率移动装置诸如利用通用处理器不用专用 DSP 或传统硬件的通用手持装置上显示视频。

本发明的另一个目的是提供一种用于无线连接的移动装置的高性能低复杂性的软件视频编解码器。无线连接可以由以下无线网络操作形式提供的, 这些网络可以通过分组交换或电路交换的按 CDMA、TDMA、FDMA 发送模型操作的 GSM、CDMA、GPRS、PHG、UMTS、IEEE 802.11 等网络。

本发明的另一个目的是发送色预量化数据, 用于当利用使用连续色表示的色彩时在客户机上以 8 比特显示的色量化实时色数据 (映射任何非固定的 3 维数据为 1 维数据)。

本发明的另一个目的是利用无另外数据额外开销或处理额外开销

的方式在单场景中支持多个任意构形视频目标。

本发明的另一个目的是无缝集成音频、视频、文本、音乐和动画图形到视频场景中。

本发明的另一个目的是在视频比特流中直接将控制信息加到各个目标上，定义在场景中各个目标的交互行为、修饰（rendering）、组合、数字权限（rights）管理信息、和压缩数据的翻译。

本发明的另一个目的是在视频和控制修饰中与个别目标的交互，以及正在显示的内容的组合。

本发明的再另一个目的是提供交互视频处理能力，能够改善单个视频目标的修饰参数，当条件变真时执行分配给各个视频目标的特定动作，并改善整个系统状态的能力并执行非线性视频导航。这是通过加到各个单个目标上的控制信息实现的。

本发明的另一个目的是提供非线性视频和组合媒体的能力，其中系统能够响应请求，通过跳到规定的场景控制用户与超链路连接的各目标的交互。在另外的例子中，采取通过视频规定部分的路径不是直接由用户与其他非直接相关目标交互确定的。例如，系统可以跟踪以前已经收视过的场景并且根据这个历史自动确定待显示的下一个场景。

在内容服务期间交互跟踪数据可以提供给服务器。为了下载内容，交互跟踪数据可以存储在装置中用于以后返回服务器的同步。在内容离线重放期间选择的超链路请求或者附加信息请求将被存储并被发送到服务器，用于在下一次同步中的执行（形式和交互事数据的异步加载）。

本发明的另一个目的是提供通过面向目标视频的相同交互控制，不论是视频数据正在从远端服务器流出还是正从本地存储器离线重放。这允许交互视频的应用在下列分配的各种情况下：流动（“拉”）、预定（“推”）、和下载。当利用下载或预定分配模型时，提供从客户装置自动和异步的形式和交互数据的加载。

本发明的一个目的是在一个场景中作音频/视频目标的动画修饰



参数。这包括：位置、比例、取向、深度、透明度、色彩和音量。本发明的目的在于按照用户交互的直接或间接结果，通过定义修饰参数的固定的动画路径，从远端服务器发送命令修改该修饰参数，并改变修饰参数实现这个目的，例如当用户点击一个目标时激活动画路径。

本发明的另一个目的是定义当用户与目标交互时执行的个别音频-视频目标的行为，其中这些行为包括动画、超链接、设置系统的状态/变量和动态媒体组合的控制。

本发明的另一个目的是有条件地对各个目标执行立即动画和行为动作。这些条件可以包括系统变量的状态、定时器事件、用户事件和各个目标之间的相关性（例如，重叠）、延迟这些动作直至各个条件变真的能力、和定义复杂条件表达的能力。还可能从一个目标到另一个目标重新指向任何控制，使得与一个目标的交互影响另一个目标，而非自身。

本发明的另一个目的包括产生视频菜单和寄存用户选择的简单形式的能力。所述形式是如果系统在线能自动同步地加载到远端服务器或如果系统离线能异步地加载。

本发明的一个目的是提供一种交互视频，包括包括定义各环路的能力，诸如播放单个目标的内容的环路或目标控制信息的环路或者整个场景的环路。

本发明的另一个目的是提供多信道控制，其中用户可以改变收视内容流到另外的信道，例如从/到多点传送（分组或电路交换的）信道到/从单点传送（分组交换连接的）会话。例如交互目标行为可以被用于实现信道改变的特征，其中在支持两种连接模型的装置中，与一个目标的交互通过从分组交换改变到电路交换连接执行改变信道，并在电路交换连接中在单点传送和广播信道之间的变化并且反之亦然。

本发明的另一个目的是通过动态媒体组合（“DMC”）提供内容个性化，这是正在收视的同时，允许显示视频场景的实际内容动态实时地改变，这是通过插入、去掉或取代该场景包含的任何任意构形的视频/音频目标，或者通过视频修剪改变场景。

一个例子将是含有视频目标成分的娱乐视频，该视频目标部分涉及用户的概况。例如在移动的场景中，一个房间可能含有高尔夫运动设备而不是网球设备。在广告媒体中这应当特别有用的，这里存在着一个相容的消息，但利用各种各样的可替代的视频目标部分。

本发明的另一个目的是能够利用或不利用交互行为，在图象中有针对性地将交互广告视频目标的传送和插入到收视场景，作为动态媒体处理的实施例。广告目标可以根据一天的时间、地理位置、用户的概况等定标到用户。另外，本发明的目的是允许处理对于用户与所述目标的交互操作（例如，用户的点击）立即的或者延迟的各种类型交互响应，其中包括：去掉广告、执行诸如立即以另外的目标取代该目标、或以新的场景替代收视的场景、寄存离线还要继续操作的用户或者在当前级别场景/会话结束时跳到新的超链路目的地或连接、或和改变广告目标的透明度或者使它远去或消失的 DMC 操作。在用户与广告目标交互跟踪的情况下，当存在按实时提供场景时，还允许订制预期目的或广告效果评估。

本发明的另一个目的是补贴（subsidise）与无线网相关的话费，或利用在呼叫期间或呼叫结束时自动显示用于发起人呼叫的发起人视频广告目标，通过广告的智能电话的使用。另外一种方案，如果用户与该目标执行某些交互的话，在呼叫之前、期间或以后显示一个交互视频目标，提供响应关系。

本发明的一个目的是提供一种用于移动装置的无线交互电子商务系统，该系统利用在线和离线情况的音频和视频数据。该电子商务包括市场运营/促销目的，利用超链路链接的图象广告或利用非线性导航的交互视频节目集，或者直接在线购物，其中可以产生作为目标的单独销售项，以便用户可以与它们进行交互操作，诸如将它们拖到购物篮等。

本发明的目的包括一种方法和系统，自由地提供公共的，（或补贴费用）存储器装置，诸如紧凑闪速存储器或者存储棒或具有某些其它形式因素的存储器装置，这些存储器装置含有带广告或促销内容或

产品信息的交互视频节目集。该存储器装置最好是只读装置，虽然其它类型存储器也可以使用。该存储器装置可以被配置为向生产商提供反馈机制的，可以利用在线通信、或者通过在存储卡写入某些数据，然后这些数据存放在某些收集点。不利用物理存储卡，利用该装置注意到是否准备接收数据和可接收量，随着导航注入信息到装置中，利用本地无线分配也可以达到相同的目的。

本发明的一个目的是当下载时向用户发送交互视频节目集、视频化（videozines）和视频（活动性）书等，然后使得它们可以与包含填写表格等的节目集进行交互。如果存在视频节目集并由用户操作或者交互，然后当客户机重新变为在线时，这些用户数据/形式将被异步加载到始发服务器。如果希望，加载可以自动地和/或异步地执行。这些节目集可以包含用于训练/教育、市场或促销、产品信息目的的视频，并且收集的用户信息可以是测试、调查、对更多信息的请求和购买定单等。交互视频节目集、视频化和视频（活动性）书可以利用图象广告目标的方式产生。

本发明的另一个目的是利用我们的根据交互视频方案的目标，根据用于移动装置的用户接口产生唯一的视频。

本发明的另一个目的是提供用于无线连接的移动用户的视频电子函件，其中可以产生和订制电子贺卡和消息，并在各个用户中传送。

本发明的另一个目的是提供如在运动场或诸如机场、购物商业街之类的其它本地环境的本地广播，这种业务具有返回信道交互用户的请求，用于附加信息或电子商务应用。

本发明的另一个目的是利用交互视频系统提供一种在线应用的语音命令和控制的方法。

本发明的另一个目的是提供无线超薄客户机，经无线连接接入到远端计算服务器。该远端计算服务器可以是私人拥有的计算机或者由应用服务提供商提供的计算机。

本发明还有一个目的是提供在低端无线装置上包含多方电视会议的电视会议，该电视会议带或不带图象广告。

本发明的另一个目的是提供一种视频监视方法，由此无线视频监视系统从视频摄象机，视频存储器装置，电缆电视和广播电视输入信号，在无线连接的 PDA 或移动电话上远程收视的码流互联网视频技术。本发明的另一个目的是利用街道交通摄象机提供交通监视业务。

## 发明内容

### 系统/编解码器方面

如果希望的话，本发明提供在低功率移动装置上利用软件发码流和/或运行视频的能力。本发明还提供以 4 元树为基础的编解码器，用于色映射视频数据。本发明还提供利用具有透明叶表示的以 4 元树为基础的编解码器，随同对任意构形定义的支持，利用 FIFO 的叶色彩预测，底层节点消除。

本发明还包括以 4 元树为基础的编解码器，对于任意构形定义具有对非底层叶的第  $n$  序插入和对底层叶的第 0 序插入。因此，本发明的各个实施例的特征可以包括一个或多个如下特征：

发送色量化信息到允许实时的客户机侧进行色量化；

利用动态 8 元树数据结构代表 3D 数据的映射，分隔为用于矢量量化的自适应码簿；

无缝组合集成音频、视频、文本、音乐和动画图象到一个无线流视频场景的能力；

在单一场景中支持多任意构形视频目标。这个特征是利用非特别数据额外开销或处理额外开销实现的，例如通过编码从亮度或纹理信息分出的附加构形信息；

基本文件格式结构，诸如文件实体层次、目标数据流、修饰的分别规范、定义和内容参数、目录、场景、和目标基础控制；

与无线流视频中的个别目标交互的能力；

在视频码流中加控制数据到各个目标上，控制交互行为、修饰参数、组成等的能力；

加数字正确管理信息到视频或图形动画数据流上的能力，用于以无线流为基础的分配和用于下载和重放为基础的分配。

产生视频目标用户接口（“VUI's”），代替常规的图形用户接口（GUI's）；和/或

使用 XML 为基础的标记语言（“IACML”）或类似源程序定义目标控制的能力，诸如在多媒体表达中 DMC 功能的修饰参数和程序控制。

### 交互方面

本发明还提供一种通过支持以下内容用于控制用户交互和动画（自动作）的方法和系统：

- 用于从流服务器发送目标控制修改数据内容或者修饰内容的方法和系统。
- 在数据文件中加目标控制修改数据内容或者修饰内容。
- 根据直接或间接的用户交互，客户机可以任选地执行由目标控制定义的动作。

本发明还提供一种加可执行的行为到各个目标上的能力，包括：对视频场景中的音频/视频目标的实施参数的动画、超链路、启动定时器、发出话音呼叫、动态媒体组合动作、改变系统状态（例如，暂停/播放）、改变用户变量（例如，设置布尔运算标志）。

本发明还提供一种当用户事件发生（按暂停键、或按键）时，或者当系统事件发生（例如，达到场景结束）时，当用户具体与目标交互（例如，点击目标或拖动目标）时动作目标行为的能力。

本发明还提供一种分配条件给各个动作和行为的方法和系统，这些条件包括：定时器事件（例如，定时器已终止）、用户事件（例如，按下键）、系统事件（例如，场景 2 正在播放）、交互事件（例如，用户已点击了目标）、目标之间的关系（例如，重叠）、用户变量（例如，布尔运算标志设置）、和系统状态（例如，播放或者暂停、码流播放或独立播放）。

另外,本发明利用 AND-OR 平面逻辑在动作执行前等待各条件变真提供形成复杂条件表达能力,清除等待动作的能力,重新定标与各个目标交互的结果和从一个目标到另一个目标的其它控制的能力,根据用户的交互在播放的同时允许各目标被其它各目标取代,和/或通过与现存目标的交互允许各新的目标的产生或例示。

本发明提供定义目标数据(例如,对于单个的各个目标的帧序列)、各目标控制(例如,修饰参数)、和整个场景(对于所有各个目标和控制重新开始帧序列)的环型播放的能力。

再有,本发明提供产生用于用户反馈的形式或用户控制的菜单和按码流移动视频交互的各种形式的功能,和拖各个视频目标到其它各个目标的顶端使得系统状态变化的能力。

### 动态媒体组合

本发明提供允许通过修改场景改变整个视频成分和通过修改各目标改变整个场景成分的能力。这可以在在线码流、离线播放视频(独立地)、和混合的情况下执行。单个图象中的各目标可以被另外的目标取代、可以增加到当前的场景、和从当前场景中删除。

在包括固定、自适应、和用户处于中间的3种模型下执行 DMC。对 DMC 支持的本地目标库可以被用于存储 DMC 中使用的目标,存储的目标用于直接播放,该目标可以由码流服务器进行管理(插入、更新、清除),并且该目标可以由服务器进行查询。另外,对 DMC 支持的本地目标库具有对库存目标的翻译控制、对非持久库存目标的自动期满检查、和从服务器中自动目标更新。另外,本发明包括对库存目标的多级别访问控制,对于每个库存目标支持唯一 ID、具有每个库存目标的历史或状态,并且在两个以后之间可以使能特殊媒体目标的共享。

### 另外的应用

本发明提供经无线连接接入远端计算服务器的超薄客户机,允许

用户产生、订制和发送电子贺卡给移动智能电话，使用语音命令的处理控制视频显示，利用非线性导航与服务器的用于训练/教育目的的交互码流无线视频的使用，发送码流卡通/图象动画到无线装置，无线码流交互视频电子商务应用，利用视频目标和码流视频定标图象中广告。

另外，本发明允许发送生动交通视频码流该给用户。这可以以多个可选择的方案执行，包括用户拨一个特殊电话号码并且然后选择交通摄象机位置，通过操作员/交换机收看该范围的图象，或者用户拨一个特殊电话号码和该用户的地理位置（从 GPS 或小区三角定位获得）被用于自动提供交通摄象机的选择进行收看。另外一种方案是用户可以寄存在特殊的服务器，该服务器的提供商将呼叫该用户并自动发表示可能具有潜在交通塞拥的乘车路线的视频信息。为此目的，当寄存的用户选择一个提出的路线时，并可以帮助确定该路线。在任何情况下，系统可以跟踪用户的速度和位置，确定行驶的方向和正在遵循的路线，然后沿着潜在路线搜索监视交通摄象机的表，确定是否有任何地方被拥塞。如果有拥塞，系统将呼叫驾驶员并提供交通图象。对不动的用户或者以步行速度行驶的用户将进行呼叫。另外一种方案，规定指示拥塞的交通摄象机，系统可以通过寄存用户的表进行搜索行驶在该道路上的用户并且对他们进行告警。

本发明还提供免费或者补贴费用的公共存储装置，例如紧凑闪速存储器、存储棒、或诸如光盘之类的以任何其它形式的存储装置，这些存储装置中含有具有广告或者促销内容或者产品信息的交互视频节目集。虽然如果需要可以使用诸如读/写存储器之类的其它类型的存储器，但是对于用户来说存储装置最好使用只读存储器。该存储装置可以被配置为提供对制造商的反馈机制，这种机制利用在线通信，或者利用将某些数据写回到放置在某些收集点的存储装置。

不用物理的存储卡或者其它存储装置，也可以实现这种相同的处理，即利用本地无线分配，考虑是否该装置准备接收数据，并如果准备接收数据，可接收多大数量，随着该装置的导航注入信息到该装置中。所包含的步骤包括：a) 移动装置进入本地无线网的范围（这可以

是 IEEE 802.11 或蓝牙等网络类型)，网络检测载波信号和服务器的连接请求。如果同意，客户机通过可闻性告警或某些其它方法告警用户，指示将进行传输；b) 如果该用户配置的移动装置接受这些连接请求，则建立与服务器的连接，否则该请求被拒绝；c) 客户机向服务器发送配置信息，诸如显示屏尺寸、存储器容量和 CPU 速度、装置制造商/型号和操作系统；d) 服务器接收这个信息并选择正确的数据流发送到客户机；如果都不适合，则中止连接；e) 在信息被传送以后，服务器结束该连接并且客户机告警该用户传输结束；和 f) 如果在传输完成以前，由于丢失连接不正当地结束传输，则客户机清掉使用的任何存储器并且自身重新开始新的连接请求。

#### 本发明的陈述

按照本发明，提供一种产生面向目标的交互多媒体文件的方法，包括：

分别按照视频包码流、文本包码流、音频包码流、音乐包码流、和/或图形包码流编码视频、文本、音频、和/或图形单元中的至少一个的数据；

组合所述包码流为单一自取向目标，所述目标含有其自己的控制信息；

放置多个所述目标到数据流中；和

分组单一连接的自取向场景中所述各数据流的一个或多个，所述场景包括作为各个包的序列中初始包的格式定义。

本发明还提供从非静止 3 维数据集为 1 维的实时映射的方法，包括步骤：

预计算所述数据；编码所述映射；

发送编码的映射到客户机；和

所述客户机应用所述映射到所述数据。



本发明还提供一种在面向目标交互视频系统中用于动态改变显示视频的实际内容的系统，包括：

包含交互多媒体文件格式的动态媒体组合处理，包括含有视频、文本、音频、音乐、和/或图形数据的各个目标，其中所述各目标的至少一个包括数据流，所述各数据流的至少一个包括一个场景，所述各场景的至少一个包括一个文件；

用于提供文件信息的目录数据结构；

用于使各个目标组合在一起的正确组合的选择机制；

用于根据所述目录信息利用目录信息和所述各个目标的位置知识的数据流管理器；

用于用户收视的同时，实时地在所述场景中和在所述视频中的各个场景中插入、删除、或替代所述目标的控制机制。

本发明还提供一种面向目标交互多媒体文件，包括：

一个或多个连续的自包含场景的组合；

每个所述场景包括作为第一包的场景格式定义，和接着所述第一包的一个或多个数据流的一个组；

来自所述第一数据流的每个所述数据流部分包括可以被任选地解码的各个目标并且根据按所述第一数据流中的目标控制信息规定的动态媒体组合处理进行显示；和

每个所述数据流包含一个或多个单个自含目标并且由一个终端流标志定界；所述各个目标的每个含有自身的控制信息并且由组合各个包码流形成的；通过编码未加工的交互多媒体数据形成的所述包码流包括：视频、文本、音频、音乐、或图形单元的至少一个或者组合，分别作为视频包码流、文本包码流、音频包码流、音乐包码流、音乐包码流和图形包码流。

本发明还提供一种提供低功率装置的语音命令操作的方法，该低功率装置能够操作码流视频系统，包括以下步骤：

在所述装置捕捉用户的话音；  
压缩所述话音；  
将所述压缩话音的编码样值插入到用户的控制包中；  
发送所述压缩的话音到能够处理语音命令的服务器；  
所述服务器执行自动语音识别；  
所述服务器映射该转录的话音为一个命令集；  
所述系统检查是否所述命令由所述用户或所述服务器产生；  
如果所述转录的命令是来自所述服务器，则所述服务器执行所述命令；  
如果所述转录的命令是来自所述用户，则所述系统转移所述命令到所述用户装置；  
所述用户执行所述命令。

本发明还提供一种图象处理方法，包括以下步骤：  
根据图象的颜色产生色图；  
利用该色图确定该图象的关系；和  
确定利用色图表示的该图象的至少一部分的相对运动。

本发明还提供一种确定图象的编码表示的方法，包括：  
分析表示颜色的所使用的比特数；  
当表示颜色使用的比特数超过第一值时，利用第一标志值和第一预比特数表示颜色；和  
当表示颜色使用的比特数未超过第一值时，利用第二标志值和第二预比特数表示颜色。

本发明还提供一种图象处理系统，包括：  
用于根据图象的颜色产生色图的装置；  
用于利用色图确定图象表示的装置；和  
用于确定利用色图表示的该图象的至少一部分的相对运动的装

送所述数据；

由客户机计算装置接收该图象信息；和

由客户机计算装置显示该图象信息。

本发明还提供一种远端控制计算机的系统，包括：

用于根据数据在服务器上执行计算操作的装置；

用于根据计算操作在服务器上产生图象信息的装置；

用于经无线连接从该服务器发送图象信息到客户机计算装置，而不发送所述数据的装置；

用于由客户机计算装置接收该图象信息的装置；和

用于由客户机计算装置显示该图象信息的装置。

本发明还提供一种发送电子贺卡的方法，包括以下步骤；

输入指示贺卡的特征的信息；

产生对应于该贺卡的图象信息；

编码该图象信息为具有控制信息的目标；

通过无线连接发送具有控制信息的目标；

由无线手持计算装置接收具有控制信息的目标；

由无线手持计算装置解码具有控制信息的目标；

显示在手持计算装置中已被解码的贺卡图象。

本发明还提供一种发送电子贺卡的系统，包括：

用于输入指示贺卡的特征的信息的装置；

用于产生对应于该贺卡的图象信息的装置；

用于编码该图象信息为具有控制信息的目标的装置；

用于通过无线连接发送具有控制信息的目标的装置；

用于由无线手持计算装置接收具有控制信息的目标的装置；

用于由无线手持计算装置解码具有控制信息的目标的装置；

用于显示在手持计算装置中已被解码的贺卡图象的装置。

置。

本发明还提供一种用于确定图象的编码表示的图象编码系统，包括：

用于分析表示颜色所使用的比特数的装置；

当表示颜色使用的比特数超过第一值时，利用第一标志值和第一预比特定数表示颜色的装置；和

当表示颜色使用的比特数未超过第一值时，利用第二标志值和第二预比特定数表示颜色的装置。

本发明还提供一种处理各个目标的方法，包括以下步骤：

按源程序语言分析信息；

读多个数据源，该数据源含有至少以视频、图形、动画、和音频形式的多个目标；

根据按源程序语言的信息，加控制信息到多个目标上；和  
交错该多个目标到数据流和文件的至少一个中。

本发明还提供一种处理各个目标的系统，包括：

用于按源程序语言分析信息的装置；

用于读多个数据源的装置，该数据源含有至少以视频、图形、动画、和音频形式的多个目标；

用于根据按源程序语言的信息，加控制信息到多个目标上的装置；  
和

用于交错该多个目标到数据流和文件的至少一个中的装置。

本发明还提供一种远端控制计算机的方法，包括以下步骤：

根据数据在服务器上执行计算操作；

根据计算操作在服务器上产生图象信息；

经无线连接从该服务器发送图象信息到客户机计算装置，而不发

本发明还提供一种控制计算装置的方法，该方法包括：  
由计算装置输入音频信号；  
编码该音频信号；  
发送该音频信号到远端计算装置；  
在远端计算装置翻译该音频信号并产生对应于该音频信号的信息；  
发送对应于该音频信号的信息到该计算装置；和  
利用对应于该音频信号的信息控制该计算装置。

本发明还提供一种控制计算装置的系统，该方法包括：  
由计算装置输入音频信号；  
编码该音频信号；  
发送该音频信号到远端计算装置；  
在远端计算装置翻译该音频信号并产生对应于该音频信号的信息；  
发送对应于该音频信号的信息到该计算装置；和  
利用对应于该音频信号的信息控制该计算装置。

本发明还提供一种执行发送的系统，包括：  
用于在无线手持装置上显示广告的装置；  
用于从无线手持装置发送信息的装置；和  
用于接收与由于显示广告已经发送的信息相关的打折价格的装置。

本发明还提供一种提供视频的方法，包括以下步骤：  
确定是否一个事件已经发生；和  
响应于该事件，通过该区域的视频无线传输，获得给一个用户的一个区域的发送的视频。

本发明还提供一种提供视频的系统，包括：

用于确定是否一个事件已经发生的装置；

用于获得一个区域的视频的装置；和

用于响应于该事件，通过该区域的视频无线传输，发送到一个用户的装置。

本发明还提供一种面向目标多媒体视频系统的系统，能够支持多个任意构形视频目标，不需要特别数据的额外开销和处理的额外开销，以提供视频目标构形信息。

本发明还提供一种由服务器始发通信传送多媒体内容到无线装置的方法，其中内容按希望的时间或成本效率的方式进行预先安排以便传送并且所述用户经装置的显示器或其它指示器被告警传送完成。

本发明还提供一种交互系统，其中存储的信息可以进行离线收视并存储用户输入并且当所述装置下一步连接在线时，通过无线网络将自动转移交互到特定的远端服务器。

本发明还提供一种视频编码方法，包括：

利用目标控制数据作为视频目标编码视频数据；并且

产生包含多个所述视频目标的数据流，所述视频目标具有各自的视频数据和目标控制数据。

本发明还提供一种视频编码方法，包括：

根据色减少表示，量化视频流中的色数据；

产生编码的代表所述量化的色和透明范围的视频帧数据；并且利用所述编码的视频数据产生编码的音频数据和目标控制数据。

本发明还提供一种视频编码方法，包括：

- (i) 选择对于视频数据的每个视频帧的减少的颜色集；
- (ii) 逐帧地协调颜色一致；
- (iii) 执行运动补偿；
- (iv) 根据感觉上的色差测量，确定一个帧的更新区；
- (v) 根据步骤(i)到(iv)将所述各帧的视频数据编码为各视频目标；和
- (vi) 包括每个视频目标动画、实施和动态组成控制。

本发明还提供一种无线码流视频和动画系统，包括：

- (i) 便携监视装置和第一无线通信装置；
- (ii) 用于存储压缩数字视频和计算机动画的服务器，并且能使用户从可用视频库中浏览和选择数字视频进行收视；并且
- (iii) 至少一个接口模块包含第二无线通信装置，用于从服务器向便携监视装置传送可传送数据，该便携监视装置包括用于接收所述可传送数据、变换该可传送数据为显示视频图象的视频图象、并允许用户与服务器通信交互地浏览和选择进行收视的视频的装置。

本发明还提供一种提供无线视频码流和动画的方法，至少包括以下步骤之一：

- (a) 通过广域网从远端服务器下载并存储压缩的视频和动画数据，用于以后从本地服务器的传输；
- (b) 允许用户从存储在本地服务器的视频数据库中浏览和选择数字视频，进行收视；
- (c) 发送数据到便携监视装置；和
- (d) 处理数据在便携监视装置上显示图象。

本发明还提供一种提供交互视频节目集的方法，包括以下各个步骤的至少一个：

(a) 通过以下规定产生视频节目集, (i) 在节目集中的各种场景和可能出现在每个场景中的各种视频目标, (ii) 规定预置的和用户可选择的场景导航控制和对于每个场景的单个构成规则, (iii) 对每个媒体目标规定修饰参数, (iv) 规定对媒体目标的控制, 产生各个形式, 收集用户反馈, (v) 集成压缩的媒体流和目标控制信息为成分数据流。

本发明还提供一种向移动装置产生和发送视频贺卡的方法, 包括至少一个步骤:

(a) 允许用户通过以下步骤产生视频贺卡 (i) 从一个库中选择模板视频场景或动画, (ii) 通过增加用户供给的文本或音频模板或从一个库中选择视频模板插入将作为场景中的角色订制模板;

(b) 从用户获得 (i) 识别细节, (ii) 理想的传送方法, (iii) 付费细节, (iv) 指定接收的移动装置的号码; 和

(c) 取决于指定的传送方法排队贺卡, 直至带宽变为可用或者可以获得非峰值传送, 轮询各个接收装置, 看是否能够处理贺卡, 并且如果可以处理, 则传送到指定的移动装置。

本发明还提供一种用于解码编码的数据的视频解码方法。

本发明还提供一种动态色间隔空间编码方法, 允许进一步将色量化信息发送到客户机, 能根据色减少的方式实现实时客户机。

本发明还提供一种有目的的用户和/或局部视频广告的方法。

本发明还提供一种可以是无线的和能够提供接入远端服务器的超薄客户机。

本发明还提供一种多方电视会议的方法。

本发明还提供一种动态媒体组合的方法。

本发明还提供一种允许用户订制并传送电子贺卡和明信片到移动智能电话的方法。

本发明还提供一种用于多媒体数据的无线码流的纠错方法。



本发明还提供用于分别执行上述方法之一的系统。

本发明还提供服务器软件，用于允许用户对视频数据的无线流的纠错的方法。

本发明还提供一种计算机软件，用于分别执行上述各方法的任何一个的步骤。

本发明还在需要的系统上提供视频。本发明还提供一种视频安全系统。本发明还提供一种交互移动视频系统。

本发明还提供一种处理语音命令控制视频显示器的方法。

本发明还提供包括用于控制面向目标视频和/或音频的码的软件。有利地是，该码可以包含基于 XML 的 IAVML 结构。

## 附图说明

下面仅以例子的方式参照附图描述本发明的各个优选实施例，其中：

图 1 是本发明的一个实施例的面向目标多媒体系统的简化框图；

图 2 是表示交叉组合入图 1 的实施例的面向目标数据流的 3 种主要包类型的示意图；

图 3 是表示在本发明的面向目标多媒体播放机实施例中数据处理的 3 个阶段的框图；

图 4 是表示在按照本发明的面向目标数据文件中各目标类型分层的示意图；

图 5 是表示按照本发明的数据文件或码流中典型包序列的图；

图 6 是表示按照本发明的面向目标多媒体播放机的客户机与服务器部件之间的信息流的图；

图 7 是表示按照本发明的面向目标多媒体播放机客户机的主要部件的框图；

图 8 是按照本发明的面向目标多媒体播放机客户机的功能部件的框图；

图 9 是描述按照本发明的多目标客户机分离修饰（rending）处理

主要步骤的流程图；

图 10 是按照本发明的客户机修饰引擎的优选实施例的框图；

图 11 是按照本发明的客户机交互引擎的优选实施例的框图；

图 12 是描述具有 DMC 功能的交互多目标视频场景的实施例的组成部分图；

图 13 是描述客户机执行播放按照本发明的交互面向目标视频处理中的主要步骤的流程图；

图 14 是按照本发明的交互多媒体播放机的本地服务器部分的框图；

图 15 是按照本发明的远端码流服务器的框图；

图 16 是描述由按照本发明的客户机执行的动态媒体部分的主要步骤的流程图；

图 17 是描述按照本发明的由服务器客户机执行的动态媒体部分的主要步骤的流程图；

图 18 是按照本发明的面向目标视频编码器的框图；

图 19 是由按照本发明的视频编码器执行的主要步骤的流程图；

图 20 是按照本发明的视频编码器中的输入色处理部分的框图；

图 21 是用于按照本发明的视频编码器中的范围更新选择处理的各个部分的框图；

图 22 是用于视频编码中的 3 个快速运动补偿方法的图；

图 23 是用于按照本发明的视频编码器中的树形分裂方法的图；

图 24 是编码来自按照本发明的视频压缩处理的数据结果所执行的主要步骤的流程图；

图 25 是用于编码按照本发明的色图更新信息的各步骤的流程图；

图 26 是编码对于按照本发明的正常预测帧的 4 元树结构数据的各步骤的流程图；

图 27 是编码按照本发明的 4 元树结构中叶色的步骤的流程图；

图 28 是按照本发明由视频编码器执行的压缩视频关键帧的主要步骤的流程图；

图 29 是按照本发明由视频编码器利用另外一种编码方法执行的压缩视频的主要步骤的流程图；

图 30 是按照本发明在客户机中按实时地执行实时色（矢量）量化主要包含的预量化处理的流程图；

图 31 是按照本发明的语音命令处理中的主要步骤的流程图；

图 32 是按照本发明的超薄计算客户机局域网（LAN）系统的框图；

图 33 是按照本发明的超薄计算客户机广域网（WAN）系统的框图；

图 34 是按照本发明的超薄计算客户机远端 LAN 服务器系统的框图；

图 35 是按照本发明的多方无线电视会议系统的框图；

图 36 是按照本发明的定标图象中的用户广告的交互‘按需式视频’系统的一实施例的框图；

图 37 是传送和操作处理按照本发明的图象中定标用户广告的交互的一个实施例所包括的主要步骤的流程图；

图 38 是在传送和操作处理按照本发明的交互视频节目集的一个实施例的所包括主要步骤的流程图；

图 39 是在一个交互视频节目集的实施例中的可能用户交互的序列的流程图；

图 40 是按照本发明的包含视频数据的基于推或拉分配的主要步骤的流程图；

图 41 是按照本发明的交互‘按需式视频’系统的框图，具有以远端服务器为基础的包括用户认证、接入控制、计费 and 利用率计量的数字权限管理功能；

图 42 是按照本发明播放需要的码流无线视频的处理中播放机软件执行的主要步骤的流程图；

图 43 是按照本发明的视频安全/监视系统的框图；

图 44 是按照本发明的电子贺卡系统和服务的框图；

图 45 是按照本发明产生和发送个性化电子贺卡或视频电子函件到移动电话所包括的主要步骤的流程图；

图 46 是表示利用 MPEG4 标准描述集中参数场景的框图；

图 47 是表示按照本发明提供色量化数据给解码器用于实时色量化的主要步骤的框图；

图 48 是表示按照本发明的目标库的主要部分的框图；

图 49 是表示按照本发明的视频解码器的主要步骤的流程图；

图 50 是按照本发明解码 4 元树形编码的视频帧包括的主要步骤的流程图；

图 51 是按照本发明的解码 4 元树的叶色中包括的主要步骤的流程图。

## 本发明详细描述

### 术语表

**比特流：**从服务器向客户机发送的比特序列，但可以被存储在存储器中；

**数据流：**一个或多个交叉的包流；

**动态媒体成分：**改变按实时表现的多目标多媒体的成分；

**文件：**面向目标多媒体文件；

**图象中目标：**在场景中重叠的视频目标；

**媒体目标：**一个或多个交叉的媒体类型的组合，包括：音频、视频、矢量图形、文本和音乐；

**目标：**一个或多个交叉的媒体类型的组合，包括：音频、视频、矢量图形、文本和音乐；

**包流：**属于从服务器向客户机发送的一个目标的各个数据包序列，但可以被存储在存储器中的；

**场景：**一个或多个码流的封装，包括多目标多媒体表现；

**码流：**一个或多个交叉的包流的组合，存储在一个面向目标多媒

体文件中；

**视频目标：**一个或多个交叉的媒体类型的组合；包括音频、视频、矢量图形、文本和音乐。

## 缩写

**FIFO：**先进先出缓冲器；

**IAVML：**交互音频可视标记语言；

**PDA：**个人数字助理；

**DMC：**动态媒体组合；

**IME：**交互管理引擎；

**DRM：**数字权限管理；

**ASR：**自动语音识别；

**PCMCIA：**个人计算机存储卡国际协会。

## 一般系统结构体系

描述在这里的处理和算法形成一种可实现的技术平台，用于诸如电子商务之类的先进增强式交互多媒体应用。所描述的方法的最大优点是如果希望的话，它们可以在诸如移动电话和 PDA 之类的非常低的处理功率的装置上仅利用软件执行。这将从如图 42 所示的流程图和伴随的描述中变得更明显。对于这种技术，特定的视频编解码器是基本的，因为它能在低功率移动视频系统中提供先进的面向目标交互处理能力。该系统的一个重要优点是在这种系统中存在低的额外开销。这种先进的面向目标交互处理比以前在无线装置能提供功能、用户体验和应用上的更新的级别。

诸如 MPEG1/2, H.263 播放机之类的典型的视频播放机给用户一种消极的经历。它们读出单一的压缩视频数据流并通过对接收数据执行单一、固定的解码转换进行播放。相反，作为描述在这里的面向目标视频播放机提供先进的交互视频能力并且允许来自多个源的多个视频目标的动态组合，订制用户体验的内容。该系统不仅允许多个、任

意构形视频目标共存，而且还根据用户的交互或者预先确定的设置，确定哪些目标可以在任何瞬间实时地共存。例如，取决于用户的爱好或用户的交互操作，视频的一个场景可以被编辑为在一个场景中作不同事情的两个不同角色中的一个。

为了提供这种灵活性，已经开发出一种面向目标视频系统，包括编码阶段、播放机客户机和服务器，如图 1 所示。编码阶段包括编码器 50，它压缩未加工的多媒体目标数据 51 为压缩的目标数据文件 52。服务器部分包括可编程、动态媒体组合部分 76，根据一个规定的原本，它将来自多个编码阶段的压缩的目标数据与定义和控制数据复用在一起，并发送产生的数据流到播放机客户机。播放机客户机包括解码引擎 62，它解压目标数据流并在发送它们到适当的硬件输出装置 61 之前修饰各种目标。

参照图 2，解码引擎 62 对 3 个交叉的数据流执行操作；压缩数据包 64、定义数据包 66 和目标控制包 68。压缩数据包 64 含有将由适当的编码器/解码器（‘编解码器’）解码的压缩目标（例如，视频）数据。在下面讨论用于编码和解码视频数据的方法。定义数据包 66 传送用于翻译压缩的数据包 64 的媒体格式和其它信息。目标控制包 68 定义目标行为、修饰、动画和交互参数。

图 3 是表示面向目标多媒体播放机中数据处理三个阶段的框图。如所示，对面向目标数据施加分别的变换，经系统显示器 70 和音频子系统产生最后的音频-视频表示。‘动态媒体组合’（DMC）处理 76 修改数据流的实际内容和发送它到解码引擎 62。在解码引擎 62 中，正常解码处理 72 提取压缩的音频和视频数据，并发送它到修饰引擎 74，在此施加其它的变换，包括：对单个目标的修饰参数的几何变换（例如，转换）。每个变换通过插入到数据流中的各个参数单个地进行控制。

最后 2 个变换的每个的特殊性质取决于动态媒体组合处理 76 的输出，因为这个输出确定传送到解码引擎 62 的数据流的内容。例如，动态媒体组合处理 76 可以插入特定视频目标到比特流中。在这种情况下

下，除了视频数据将被解码外，数据比特流将含有用于解码处理 72 和修饰引擎 74 的配置参数。

面向目标比特流数据格式允许不同种类媒体目标之间的无缝集成，支持用户与这些目标的交互，并且能对显示场景中的内容，无论来自远端服务器的还是接入本地存储内容的都可以进行编程控制。

图 4 是表示面向目标多媒体数据文件中的各个目标类型的分层的示意图。该数据格式按如下定义了各个实体的分层：可以包含一个或多个场景 81 的面向目标数据文件 80。每个场景可以包含一个或多个含有一个或多个独立同时的媒体目标 52 的流 82。媒体目标 52 可以是诸如视频 83、音频 84、文本 85、矢量图象 (GRAF) 86、音乐 87 或这样一些单元的组合 89 的单一媒体单元 89。在单一场景中，上述各个媒体类型的每个的多个例子可以与其它媒体类型同时发生。每个目标 52 可以含有包封在各个数据包中的一个或多个帧 88。在场景 81 中，当出现多于一个媒体目标 52 时，交叉每个包。单一媒体目标 52 完整地自包含实体，该实体具有虚拟地非相关性。包含一个或多个定义包 66、接着的数据包 64 和任何控制包 68 的各个包序列被定义为都具有相同目标识别号。数据文件中的所有包具有相同的首端信息（基本首端），该信息规定了对应于在包中的数据类型、在序列中该包的号和该包含有的数据量（包尺寸）的目标。在下面部分描述文件格式的细节。

与 MPEG4 系统的区别将是很明显的。参照图 46，对于各个场景 (BIFS) 01a，MPEG4 依赖于按二进制形式的集中参数场景描述，它是各节点的分层结构，这些节点可以含有各个目标和其它信息的属性。从非常复杂的虚拟实体标记语言 (VRML) 语法直接借用 BIFS 01a。在这种方法中，集中的 BIFS 结构 01a 实际上是场景本身：在面向目标视频中是基本成分，不是目标本身。视频目标数据可以被规定使用在场景中，但是不能用作定义场景本身。这样，例如一个新的视频目标不能被引入到一个场景中，除非 BIFS 结构 01a 首次被修改为包括参考该视频数据的一个节点。BIFS 还不直接参考任何目标数据流，代

之以称为目标描述符 01b 的中间独立装置在 BIFS 01a 节点中的任何 OBJ\_ID 与含有视频数据的单元数据流 01c 之间映射。因此, MPEG 方法中, 这 3 个分别的实体 01a、01b、01c 的每个是互相依赖的, 使得如果一个目标流被复制到另外的文件中, 则它丢失任何交互性能和与其相关的任何其它控制信息。因为 MPEG 4 不是目标中心的, 它的各个数据包被称为原子, 具有包括任何类型和包尺寸信息的公共首端, 但没有目标识别符。

这里所描述的格式是非常简单的, 因为没有定义是哪个场景的中心结构。代之以, 场景是自含有的并且完全由存在于该场景的目标进行定义。每个目标也是自含有的, 具有加上的规定目标的属性和交互行为的任何控制信息。新的目标可以被复制到刚通过插入其数据到比特流中的一个场景中, 为此引入所有目标控制信息以及其压缩数据到该场景中。存在着各个媒体目标之间或各个场景之间的虚拟无相互依赖。这种方法降低了复杂性和复杂的 BIF 方法相关的存储及处理的额外开销。

在视频数据的下载和播放的情况下, 允许交互, 诸如选择哪个角色出现在场景中能力的多媒体数据的面向目标操纵, 输入数据不包括具有单 ‘角色’ 目标的单一场景, 而是可以根据用户的输入进行选择或 “组合入” 在运行时间所显示的场景中的每个场景中的一个或多个可供选择的目标数据流。因为在运行时间之前不知道场景的组合, 不可能交叉正确目标数据流到场景中。

图 5 是表示数据文件中的典型包序列的图。存储的场景 81 包括多个独立可选择的流 82、为动态媒体组合处理的候选物的每个 “角色” 目标 52, 参照图 3。在场景 81 中仅第一流 82 含有多于一个 (交叉的) 的媒体目标 52。场景 81 中的第一流 82 定义场景结构、组成的各目标及其行为。在场景 81 中的附加流 82 含有任选的目标数据流 52。在每个场景 81 的开始设置流的目录 59, 使能随机地访问每个独立的流 82。

在该比特流能够支持先进的交互视频能力和动态媒体组合的同时, 它还支持 3 个实现级别、提供各种功能级别。这些是:



1. 消极媒体：单目标、非交互播放机
2. 交互媒体：单目标、有限交互的播放机
3. 面向目标积极媒体：多目标、全交互播放机

该最简单的实现提供具有单一媒体情况并且没有交互性的消极收视经历。这是一种传统的媒体播放机，在这种播放机中用户限于播放、暂停和停止正常视频或音频的重放。

下一个实现级别通过允许定位行为的各热区定义增加了交互支持消极媒体。这是通过产生具有有限目标控制功能的矢量图形目标提供的。因此，虽然它可能依然这样出现在用户面前，但该系统不再是一个照字面意义的单一目标系统。除了主媒体目标看来是透明的外，可点击矢量图形目标也是允许的其它目标类型。这允许产生简单的交互经历，诸如非线性导航等等。

最后实现的级别定义了不限制多目标和全目标控制功能的使用，包括动画、条件事件等，并且利用在这种结构中的所有成分的实现。实际上，这种级别与前面级别之间的区别可能仅在于装饰上。

图 6 是表示在面向目标多媒体系统中的客户机与服务器成分之间的信息流（或比特流）的图。该比特流支持客户机侧与服务器侧之间的交互。客户机侧的交互是经定义的动作集支持的，这些动作可能通过用户经历产生修改的各个目标调用，但这里表示为目标控制包 68。服务器侧交互支持是用户交互，表示在这里的是的用户控制包 69，经返回信道从客户机 20 中继到远端服务器 21，并且对在线用户在动态媒体组合的形式上起支配地位地提供服务/内容补充的调解。因此，处理比特流的交互媒体播放机具有客户机-服务器结构。客户机 20 响应于从服务器 21 发送给它的解码的压缩数据包 64、定义包 66 和目标控制包 68。附加地，客户机 20 可响应于目标同步，施加修饰变换、组成最后的显示输出、管理用户的输入和传送用户控制返回到服务器 21。服务器 21 可响应于来自正确源（各个源）的管理、读出、和分析部分比特流，根据来自客户机 20 的用户的具有适当控制指令的输入构成复合比特流，并且传送该比特流到客户机 20，用于解码和修饰。表

示在图 3 的部分 76 的这种服务器侧的动态媒体组合允许媒体的内容根据用户的交互或按存储的源程序预定义的设置被实时地组合。

当播放返回数据本地存储时，并还当数据是来自服务器 21 的码流时，媒体播放机支持服务器侧和客户机侧两者的交互性/功能性。因为执行 DMC 和管理源的服务器部分 21 的可响应性，在本地重放的情况下，服务器是与客户机 20 共同在一个位置上的，虽然在码流情况下是在远端位置上的。在客户机 20 访问来自本地和远端位置的源/服务器 21 数据的情况下，还支持混合操作。

### 交互客户机

图 7 是表示面向目标多媒体播放机客户机 20 的主要部分的框图。面向目标多媒体播放机客户机 20 能接收和解码由服务器 21 发送并由图 3 的 DMC 处理 76 产生的数据。面向目标多媒体播放机客户机 20 还包括多个执行解码处理的部分。与编码处理比较，解码处理的步骤是非常简单的，并且可以完全在诸如 Palm Pilot IIIc 或智能电话之类的低功率移动计算装置中由软件编辑执行。输入数据缓冲器 30 被用于保持从服务器 21 输入的数据，直至全部包已经被接收或读出。然后，直接或经由解密单元 34，数据被传送到输入数据交换/去复用器 32。输入数据交换/去复用器 32 确定要求哪个子处理器 33、38、40、42 解码该数据，并且然后根据执行子处理的包类型，传送该数据到正确的部件。独立的部件 33、38 和 42 分别执行矢量图形、视频和音频解码。解码器中的视频和音频解码模块 38 和 42 独立地解压任何发送给它们的数据并且在暂时缓冲器中执行初步的修饰。目标管理部分 40 提取对于控制视频场景使用的目标行为和修饰信息。视频显示部分 44 在从矢量图形解码器 33、视频解码器 38 和目标管理部分 40 接收的数据的基础上修饰视频目标。音频播放部分 46 在从音频解码和目标管理部分 40 接收数据的基础上产生音频。以后用户输入/控制部分 48 产生指令并控制由显示和重放部分 44 和 46 产生的视频和音频。以后用户控制部分 48 还发送控制消息返回到服务器 21。

图 8 是表示面向目标多媒体播放机客户机 20 的各个功能部分的框图，包括如下：

1. 用于主数据路径的具有任选目标存储器 39 的解码器 43（图 7 的多个部分 33、38 和 42 的组合）
2. 修饰引擎（图 7 的部分 44 和 46）
3. 交互管理引擎 41（图 7 的部分 40 和 48）
4. 目标控制 40 路径（图 7 的部分 40 的一部分）
5. 输入数据缓冲器 30 和输入数据交换/去复用器 32
6. 任选数字权限管理（DRM）引擎 45
7. 持久本地目标库 75

通过客户机系统 20 有两个主要的数据流。来自服务器 21 或持续本地目标库 75 的压缩的目标数据 52 被传送到客户机输入缓冲器 30。输入数据交换/去复用器 32 分裂经缓冲的压缩目标数据 52 为压缩数据包 64、定义包 66 和目标控制包 68。压缩数据包 64 和定义包 66 根据在包首端所识别的包类型被分别择径到适当的解码器 43。目标控制包 68 被发送到目标控制部分 40 进行解码。另外一种方案，如果目标控制包被接收规定了库更新信息，则压缩数据包 64、定义包 66 和目标控制包 68 可以从输入数据交换/去复用器 32 进行择径到用于持久本地存储的目标库 75。对于每个媒体目标和每个媒体类型，存在一个解码器 43 和一个目标存储器 39。因此不仅存在对于每个媒体类型的不同解码器 43，而且如果在场景中有 3 个视频目标，则将存在 3 个视频解码器 43 的情况。每个解码器 43 接收发送给它的适当压缩数据包 64 和定义包 66 并且在目标数据存储器 39 中缓冲解码数据。每个目标存储器 39 可响应于与修饰引擎 74 相连接的每个媒体目标的同步管理，如果解码滞后于（视频）帧更新速率，然后指令解码器 43 适当卸下各个帧。由修饰引擎 74 读出目标存储器 39 中的数据，组成最后显示的场景。对目标存储器 39 的读和写的访问是异步的，以便解码器 43 可以仅以低速率更新目标存储器 39，同时修饰引擎 74 可以以较快的速率读出数据，或者反之亦然，这取决于全面媒体同步请求。修饰引擎

74从每个目标存储器 39 读出数据并且根据来自交互管理引擎 41 的修饰信息组成最后的显示场景和声音场景两者。这种处理的结果是通过系统图形用户接口 73 处理的一系列位映像, 将被显示在显示装置 70 上和一系列音频样值, 将被传送到系统的音频装置 72。

第二数据浪通过客户机系统 20 经图形用户接口 73 从用户以用户事件 47 的形式输入到交互管理引擎 41, 在这里该用户事件被分裂, 它们的某些部分以修饰参数的形式被传送到修饰引擎 74, 并且其余部分作为用户控制包 69 通过返回信道被传送到服务器 21, 服务器 21 利用这些数据控制动态媒体组合引擎 76。为了判断在哪里或者是否用户事件被传送到系统的其它部分, 交互管理引擎 41 可以要求修饰引擎 74 执行命中测试。交互管理引擎 41 的操作由接收从服务器 21 发送的指令(目标控制包 68)的目标控制部分 40 进行控制, 该指令定义交互管理引擎 41 如何翻译来自图形用户接口 73 的用户事件 47, 和哪个动画和交互行为是与单个媒体目标相关。交互管理引擎 41 负责控制修饰引擎 74 进行修饰变换。另外, 交互管理引擎 41 负责控制目标库 75, 择径库目标到达输入数据交换/去复用器 32。

修饰引擎 74 具有如图 10 所示的 4 个主要部分。位映像编排器 35 从视频目标存储缓冲器 53 读出各个位映像并编排它们到最后显示的场景屏面 71。矢量图形原始扫描变换器 36 修饰来自矢量图象解码器的矢量图形显示表 54 为显示场景屏面 71。音频混合器 37 读音频目标存储器 55 并在传送的结果到音频装置 72 之前混合音频数据。读出各种目标存储缓冲器 53 到 55 的的序列以及如何将其内容变换为显示场景屏面 71 通过来自交互管理引擎 41 的修饰参数 56 确定。可能的变换包括: Z-序、3D 取向、位置、比例、透明度、颜色和音量。为了加速修饰处理, 可以不需要修饰整个显示场景, 而仅修饰其一部分。修饰引擎的第 4 个主要部分是命中测试器 31, 它按照交互管理引擎 41 的用户事件控制器 41c 的控制下, 对用户笔事件执行目标命中测试。

当用户通过点击或拖动一个可拖的目标选择一个键时, 和当更新动画时, 每当按照同步信息从服务器 21 接收视频数据的情况下, 显示

场景应当进行修饰。为了修饰场景，可以编排到画面外缓冲器（显示场景屏面 71），并然后移到输出装置 70。目标修饰/位映像组合处理表示在从步骤 s101 开始的图 9 中。维持包含一个指针的表，指示含有各个视频目标的每个媒体目标存储。在步骤 s102 按照 Z 序存储该表。接下来，在步骤 s103，位映像组合器得到具有最低 Z 序的媒体目标。如果在步骤 s104，再没有别的目标进行组合，则在步骤 s118，结束视频目标的修饰处理。否则，始终处于第一目标的情况下，在步骤 s105，从目标缓冲器读出解码的位映像。在步骤 s106，如果存在目标修饰控制，则在步骤 s107 设置屏幕位置、取向和比例。具体地，目标修饰控制定义适当 2/3D 几何变换，确定映射到哪个座标的目标像素。在步骤 s108，从目标缓冲器读出第一像素，并且如果在步骤 S109 有更多像素处理，则在步骤 s110，从目标缓冲器读出下一个像素。在目标缓冲器中的每个像素被单个地进行处理。在步骤 S111，如果像素是透明的（像素值是 0xFE），则修饰处理忽略该像素并且返回到步骤 s109 开始处理目标缓冲器中的下一个像素。否则，在步骤 s112，如果该像素未改变（像素值是 0xFF），则在步骤 s113，背景色像素被移到显示场景屏面。但是，在步骤 s114，如果像素既不透明又不是未改变，并且不能进行 $\alpha$ 混合，则在步骤 s115，目标色像素被移到显示场景屏面。在步骤 s114，如果能进行 $\alpha$ 混合，则执行 $\alpha$ 混合组合处理，对目标设置定义的透明度级别。但是，不像传统 $\alpha$ 混合处理，需要分别对位映像中的每个像素编码混合系数，这种方法不使用 $\alpha$ 信道。代之以，在实际位映像表示中，使用单个 $\alpha$ 值规定与插入透明范围指示相关的整个位映像的不透明度。因此，当在步骤 s116 计算新的 $\alpha$ 混合目标像素色时，在步骤 s117，该目标像素色被移到显示场景屏面。这终止对每个单独像素的处理，因此控制返回到步骤 s109，以开始目标缓冲器中新的像素的处理。在步骤 s109，如果没有像素仍将被处理，则处理返回到步骤 s104，开始处理器下一个目标。位映像编排器 35 读按照与每个媒体目标相关的 Z 序序列存储的每个视频目标，并且复制它到显示场景屏面 71。如果没有 Z 序已被明确地分配给各个目标，对于一个目

标的 Z 序值可以取与 object\_ID 相同的值。如果两个目标具有相同 Z 序，则它们被移到上阶目标 ID 的序。

如所述，位映像编排器 35 使用视频帧可以具有的 3 个区类型：将修饰色像素的区域、将使透明的区域和保持不变的区域。色像素被适当地 $\alpha$ 混合到显示场景屏面 71，并且未改变的像素被忽略，使得显示场景屏面 71 不受影响。透明像素迫使对应背景显示场景将被刷新。当正在讨论的目标的像素与某些其它目标的像素重叠时，通过简单地什么也不要作，就可以执行这种操作，但是如果该像素正在通过场景背景被直接移动，则该像素需要被设置为场景背景色。

如果目标存储器含有显示表代替位映像，则对显示表中的每个座标施加几何变换，并且在显示表中规定的图形基本要素的扫描变换期间执行 $\alpha$ 混合。

参照图 10，位映像编排器 35 支持具有不同色分辨率的显示场景屏面，并管理具有不同比特深度的位映像。如果显示场景屏面 71 具有 15、16 或 24 比特的深度，和位映像是映射 8 比特图象的色，则位映像编排器 35 从位映像中读出每个色索引值，在与特定目标存储器相关的色图中查寻该色，并且以正确形式写该色的红、绿和兰分量到显示场景屏面 71。如果该位映像是连续色调图象，位映像编排器 35 简单地复制每个像素的色值到显示场景屏面 71 的正确位置。如果显示场景屏面 71 具有 8 比特深度和色查寻表，则取决于显示的目标数适当的取值。如果仅一个视频目标被显示，则它的色图直接被复制到显示场景屏面 71 的色图。如果存在多个视频目标，则显示场景屏面 71 将被建立一个通用色图，并且设置在显示场景屏面 71 中的像素值将最接近地匹配于由位映像中的索引值指示的色。

当用户在屏幕上通过比较笔事件位置座标与每个显示的目标，已经选择一个视频目标时，修饰引擎 74 的命中测试部分 31 负责评估。如图 10 所示，由交互管理引擎 41 的用户事件控制器 41c 请求这种“命中测试”并且使用由位映像编排器 35 和矢量图象基本要素扫描变换器 36 部分提供的目标位置和变换信息。命中测试部分 31 对每个

目标施加笔事件位置的反向几何变换，并且按产生的反向变换座标的结果评估该位映像的透明度。如果该评估是真，则寄存一个命中，并且该结果返回到交互管理引擎 41 的用户事件控制器 41c。

修饰引擎的音频混合器部分 37 以循环的方式读存储在相关音频目标存储器中的每个音频帧，并根据由交互引擎提供的修饰参数 56 混合该音频数据，获得组合帧。例如，用于音频混合的修饰参数可以包括音量控制。然后，音频混合部分 37 传送混合的音频数据到音频输出装置 72。

图 8 的目标控制部分 40 基本上是一个编解码器，它从交换/去复用器输入的流中读编码的目标控制包和发指示控制的指令到交互管理引擎 41。控制指令可以被发出，改变单个目标或系统宽度的属性。这些控制是宽范围的，并包括修饰参数、动画路径的定义、产生条件事件、控制包含从目标库 75 插入各目标的媒体播放序列、分配超链路、设置定时器、设置和复位系统状态寄存器等，并且定义用户激活的目标行为。

交互引擎 41 管理多个不同的处理；图 13 的流程表示交互客户机在执行交互面向目标视频中的主要步骤。处理在步骤 s201 开始。在步骤 s202，从图 8 的目标存储器 39 或者图 8 的目标控制部分 40 的输入数据源读出数据包和控制包。在步骤 s203，如果包是数据包，则在步骤 s204 进行帧解码和缓冲。但是，如果包是目标控制包，则在步骤 s206，交互引擎 41 对目标加适当的动作。然后，在步骤 s205 对目标进行修饰。在步骤 S207，如果没有用户与目标进行交互（即，没有用户点击目标），并且在步骤 s208 没有目标等待动作，则处理返回步骤 s202，并在步骤 s202 从输入数据源读出新的包。但是，在步骤 s208 如果目标等待动作，或者没有用户交互，但在步骤 s209 目标附有动作，在步骤 s210 对动作条件进行测试，并且如果条件满足，则在步骤 S211 执行该动作。否则，在步骤 S202 从输入数据源读出下一个包。

交互引擎 41 不预测行为：交互引擎 41 可以执行或响应的所有动作和条件是由目标控制包 68 定义的，如图 8 所示。交互引擎 41 可以：

无条件地立即执行预定动作（诸如到达场景中的最后视频帧时跳回到场景的开始），或者延迟执行直至某些系统条件被满足（诸如定时器事件），或者它可以按规定的行为响应用户的输入（诸如点击或拖一个目标），这种响应可以是无条件的或者是从属于系统条件的。可能的动作包括：修饰属性变化、动画、循环地和非连续播放序列、跳到超链路、显示的目标流被可能来自持久本地目标库 75 的另外目标取代的动态媒体组合、和当规定条件或用户事件变真时调用的其它系统行为。

交互引擎 41 包括 3 个主要部分：交互控制部分 41a、等待动作管理器 41d、和动画管理器 41b，如图 11 所示。动画管理器 41b 包括：交互控制部分 41a 和动画路径插补器（interpolator）/动画表 41b，并存储当前进行中的所有动画。对每个动作动画，在由目标控制逻辑 63 规定的间隔，该管理器内插发送到修饰引擎 74 的修饰参数 56。当一个动画已经完成时，被从动画表 41b 的动作的动画表中去除，除非它被定义为循环动画。等待动作管理器 41d 包括：交互控制部分 41d 和等待动作表 41d，并存储将施加到条件变真的所有目标控制动作。交互控制部分 41a 有规律地轮询等待动作管理器 41d 并评估与每个等待动作相关的条件。如果满足对一个动作的条件，则交互控制部分 41a 将执行该动作并且从等待动作表 41d 中将其清除，除非该动作已经变定义为目标行为，在这种情况下它仍然在等待动作表 41d 上用于将来进一步执行。对于条件评估，交互管理引擎 41 使用条件评估器 41f 和状态标志寄存器 41e。交互控制部分 41a 更新状态标志寄存器 41e，并保持一组用户可定义系统标志。条件评估器 41f 在每个目标的基础上按交互控制部分 41a 发的指令执行条件评估，比较当前系统状态和状态标志寄存器 41e 中的系统标志，并且如果适当的系统标志被设置，则条件评估器 41f 通知交互控制部分 41a 该条件是真，并且将执行该动作。如果该客户机是离线的（即，没有连接到远端服务器上），则交互控制部分 41a 保持执行的所有交互动作的记录（用户事件等）。这些是暂时存储在历史/形式存储器 41d，并且当该客户机在线时利用



用户控制包 69 发送到服务器。

目标控制包 68 并因此目标控制逻辑 63 可以设置许多用户可定义系统标志。这些标志用于允许系统具有其当前状态的存储器，并且这些标志被存储在状态标志寄存器 41e 中。例如，当重放视频中的某场景或帧时，或当用户与一个目标交互时，可以设置这些标志之一。利用用户事件控制器 41c 从图形用户接口 73 接收输入的用户事件 47 监视用户交互。另外，用户事件控制器 41c 可以请求修饰引擎 74 利用修饰引擎的命中测试器 31 执行 ‘命中测试’。一般，对诸如用户笔点击/选择之类的用户笔事件请求命中测试。用户事件控制器 41c 传送用户事件到交互控制部分 41a。然后，这可以用于确定在下一个非线性视频中播放哪个场景，或者在一个场景中修饰哪些目标。在电子商务应用中，用户可以拖一个或多个图符到购物篮目标中。然后，这将寄存想要的购买。当点击购物篮时，视频将跳到检验场景，在那里显示拖到购物篮中的所有目标，允许用户确认或删除项目。分别的视频目标可以被用作按钮，指示用户希望寄存的购买单或者删除它。

目标控制包 68 并因此目标控制逻辑 63 可以包含满足任何要执行的规定动作的条件，这些条件是由条件评估器 41f 进行评估的。这些条件可以包括：系统状态、本地或码流重放、系统事件、规定的用户与目标的交互等。一个条件可以具有等待标志设置，指示该条件当前不满足，然后等待直至满足。等待标志经常用于等待诸如用户笔出现之类的用户事件。当满足一个等待动作时，从与一个目标相关的等待动作表 41d 中去除。如果设置目标控制包 68 的行为标志，则在等待动作表 41d 中将保持一个目标的动作，即使在该目标已经执行以后。

目标控制包 68 并因此目标控制逻辑 63 可以规定影响另外的目标的动作。在这种情况下，条件将对在基本首端规定的目标上被满足，而该动作是在其它目标上执行的。目标控制逻辑可以规定被传送到目标库 75 的目标库控制 58。例如，具有用户对要求的目标的点击事件的条件，并由用户事件控制器 41c 结合命中测试器 31 进行评估，并且系统在执行该指令之前将等待其变为真，目标控制逻辑 63 可以规定将

与动画一起执行的跳跃动作（超链路）。在这种情况下，动作或控制将在等待动作表 41d 中等待，直至它被执行并然后它将被去除。例如，类似这种控制可以与视频中由角色穿着的一双运动鞋相联系，这样当用户点击它们时，鞋子可以围绕场景移动并在用户改变视频几秒钟之前以在线操作提供出售鞋子信息和鞋子的购买或出价的机会。

图 12 表示多目标交互视频场景的组成。最后场景 90 包括背景视频目标 91、3 个任意构形 “信道改变” 视频目标 92、和 3 个 “信道” 视频目标 93a、93b、93c。目标可以利用用户点击事件的条件，通过分配 “行为”、“跳跃” 和 “其它” 特性的控制，定义为 “信道改变” 92。这种控制被存储在等待动作表 41d，直至场景发生的结束并且只要被点击将使得 DMC 改变场景 90 的组成。在这个说明中 “信道改变” 目标将显示正在显示在其它信道的内容的缩小形式。

目标控制包 68 并因此目标控制逻辑 63 可以具有动画标志设置，指示将跟随的多命令而不是单命令（诸如，移动到）。如果动画标志不能设置，则只要条件满足，动作就执行。随着经常修饰变化的发生，显示场景也将被更新。不像大多数修饰动作是由用户事件 47 或目标控制逻辑 63 驱动的，动画将迫使修饰自更新。动画进行更新以后，并如果整个动画完成，则动画将从动画表 41b 中去掉。动画路径插补器 41b 确定动画当前位于哪两个控制点之间。沿着动画已在两个控制点之间（‘中间’ 值）传播率的这个信息用于内插相关的修饰参数 56。该中间值被表示为根据分子与分母的比：

$$X=x[\text{开始}]+(x[\text{结束}]-x[\text{开始}]) * \text{分子} / \text{分母}$$

如果动画被设置为循环，则动画的开始时间被设置为动画结束时的当前时间，使得更新以后也不被去掉。

客户机支持如下高级别用户交互的类型：点击、拖、重叠、和移动。一个目标可以具有与其相关的按钮图象，当笔保持在一个目标之上时，这个图象被显示。如果笔压向一个目标并在规定像素上移动时，

则该目标被拖动（只要拖动没有被目标或场景保护）。拖动实际上移动了笔下的目标。当笔被释放时，该目标被移动到新的位置，除非移动被目标或场景保护。如果移动被保护，当笔被释放时，拖动的目标移回原始位置。可以实现拖动，使得用户可以下拉在其它各目标的顶上的目标（例如，拖一个项到购物篮）中。如果笔被释放，同时该笔还在其它目标之上，则这些目标被通知为与所拖目标的重叠事件。

通过目标控制包 68 在透明或深度上，各目标可以从点击、移动、拖动、或改变进行保护。在目标控制包 68 中的 PROTECT（保护）命令可以具有单个的目标范围或系统范围。如果具有系统范围，则所有目标受到 PROTECT 命令的影响。系统范围保护超越目标范围保护。

JUMPTO（跳跃）命令具有 4 种变量。一种变量允许跳跃到由超链路规定的另外的文件中新规定场景。另一种允许利用来自超链路规定的分别的文件或场景的另外媒体目标替代在当前场景中的当前播放的媒体目标流，和另外两个变量允许跳到相同文件中新的场景或利用由目录索引规定的相同场景中另外目标替代播放媒体目标。每个变量可以利用或不利用目标映射进行调用。另外，JUMPTO 命令可以利用来自本地持久目标库 75 存储的媒体目标取代当前播放媒体目标流。

虽然大多数交互控制功能可以由客户机 20 利用修饰引擎 74 结合交互管理器 41 进行处理，但某些控制例可能需要按较低级别进行处理，并被传回到服务器 21。这包括用于非线性导航的命令，以便跳到超链路和动态场景组合，除指令来自目标库 75 的目标插入的命令外。

图 8 的目标库 75 是持久的本地媒体目标库。通过称为目标库控制包的特殊的目标控制包 68 和具有目标库模型字段设置的场景定义包 66，各个目标可以插入或从这个库去除。目标库控制包定义将对目标执行的动作，包括；插入、更新、拖动和查询目标库。如果定义适当目标库动作（例如插入或更新），输入数据交换/去复用 32 可以择径压缩数据包 52 直接到目标库 75。如图 48 的框图所示，每个目标被作为分别的流存储在目标库数据存储器 75g，该库不支持多交叉目标，因为寻址是根据流号码的库 ID 的。因此，库可以包含高达 200 个分

别的用户目标，并且目标库可以利用特定场景号（例如 250）进行参考。该库还支持达 55 个系统目标，诸如缺省按钮、检查框、形式等。该库支持垃圾收集，诸如在某个时间周期以后一个目标可能被设置为过期的，此时该目标被从库中清除出。对于每个目标/流，包含在目标库控制包中的信息由客户机 20 存储，含有用于流/目标的附加信息，包括：库 id 75a、版本信息 75b、目标持续信息 75c、访问限制信息 75d、唯一目标识别符 75e 和其它状态信息 75f。该目标流附加地包括压缩目标数据 52。目标库 75 可以按照目标控制部分 40 的操纵，由图 8 的交互管理引擎 41 进行查询。这是通过对库 75 中的所有目标连续读出和比较目标识别符值找出与馈送的搜索密钥的匹配值实现的。库查询结果 75i 被返回到交互管理引擎 41，进行处理或者发送到服务器 21。目标库管理器 75h 负责管理与该目标库的所有交互。

## 服务器软件

服务器系统 21 的目的是：（i）产生正确数据流，用于客户机进行解码和修饰；（ii）通过包括 TDMA、FDMA、或 CDMA 系统的无线信道可靠地发送所述数据到客户机；和（iii）处理用户交互。数据流的内容是动态媒体组合处理 76 和由非线性媒体导航施加的非连续访问请求的函数。客户机 20 和服务器 21 两者包括在 DMC 处理 76 中。组合数据流的源数据可以来自单一源或来自多个源。在单一源的情况下，该源应当包含对组成最后数据流所要求的所有任选数据成分。因此，这个源可能包含不同场景的库，和将被用于组合的各种媒体目标的多个数据流。因为这些媒体目标可以被同时组合入一个场景，对服务器 21 的这部分提供先进的非连续访问能力，从每个媒体目标流中选择合适的的数据部分，以便交叉它们到最后的组合数据流中发送到客户机 20。在多源的情况下，将被用于组合的不同媒体目标的每个可以具有单个的源。具有按分别源的一个场景的各个组成目标减轻服务器 21 的复杂访问请求，因为虽然存在着多个源进行管理，但每个源仅需要顺序地进行访问。

支持两种源的情况。对于下载和播放功能来说，最好是传送含有打包内容的一个文件，而不是多个数据文件。对于流的播放，最好是保持各个源的独立，因为这样允许在组合处理中的更多灵活性并订制诸如定标用户的广告之类的特殊用户需要。分别源的情况还降低服务器设备的负荷，因为所有文件的访问有顺序的。

图 14 是播放本地存储文件的交互媒体播放机的本地服务器部分的框图。如图 14 所示，独立的播放机需要本地客户机系统 20 和本地单一源服务器系统 23。

如图 15 所示，码流播放机需要本地客户机 20 和远端多源服务器 24。但是，播放机还能同时播放本地文件和码流内容，这样客户机系统 20 还能从本地服务器和远端服务器两者同时接收数据。本地服务器 23 或者远端服务器 24 可以构成服务器 21。

参照具有图 14 的消极媒体播放机的最简单的情况，本地服务器 23 打开面向目标数据文件 80 和连续地读它的内容，传送数据 64 到客户机 20。当按用户控制包 68 执行用户命令时，文件的读操作可以从它的当前位置上停止、暂停、和继续，或者可以从面向目标数据文件 80 的开始端重新开始。服务器 23 执行 2 个功能：访问面向目标数据文件 80 和控制这种访问。这些可以被综合到复用器/数据源管理器 25 和动态媒体组合引擎 76 中。

在具有本地视频重放和动态媒体组合的更先进的情况中(图 14)，对于客户机不可能只顺序地读具有复用目标的一个预定流，因为当产生面向目标数据文件 80 时，复用流的内容是不知道的。因此，面向目标数据文件 80 包括用于连续存储的每个场景的多个流。本地服务器 23 随机地访问一个场景中的每个流并选择将要发送到客户机 20 用于修饰的需要的目标。另外，持久目标库 75 由客户机 20 进行维护并且当在线时可以从远端服务器进行管理。这被用作存储公共下载的各个目标，诸如用于形成检查框图象。

图 14 的数据源管理器/复用器 25 随机地访问面向目标数据文件 80，从该各种流中读出数据和控制包，用于组成显示场景，并且复用

这些在一起产生客户机 20 用于修饰组成的场景的复合包流 64。一个流纯粹是概念上的，因为没有一个流的指示开始的包。但是，存在一个流结束的包，划分流边界的界线，如图 5 的 53 所示。一般，在一个场景中的第一流含有场景中各个目标的描述。该场景中的目标控制包可以改变一个特定目标到不同流的源数据。然后，当执行本地重放时，服务器 23 需要从一个面向目标数据文件 80 中同时读出多于一个的流。并不是产生分别的线索，而是可以产生一个阵列或链接的流表。数据源管理器/复用器 25 以循环的方式从每个流中读出一个包。至少，每个流需要存储在文件中的当前位置和一个参考目标的表。

在这种情况下，当从客户机 20 接收到用户控制信息 68 时，图 14 的动态媒体组合引擎 76 选择将被组合在一起的各个目标的正确的组合，并保证根据由复用器/数据源管理器 25 提供给动态媒体组合引擎 76 目录信息，复用器/数据源管理器 25 知道到哪里找到这些目标。这还可以要求一种目标映射功能，映射具有运行时间目标识别符的存储目标识别符，因为取决于组合它们可以不同。典型的情况是，这可能发生在文件 80 中的多个场景可能希望共享一个特定视频或音频目标时。因为一个文件可以含有多个场景，这可以通过在特定“库”场景中存储共享的内容来实现。在一个场景中的各个目标具有从 0-200 范围的目标 ID，并且每次遇到新场景定义包，没有目标的场景被复位。每个包含有基本首端，规定包的类型以及参考目标的目标 ID。254 的目标 ID 代表场景，而 255 的目标 ID 代表文件。当多个场景共享一个目标数据流时，并不知道哪个目标 ID 将已被分配给不同的场景，因此，不可能在共享目标流中预选目标 ID，可能这些 ID 已经在场景中分配了。解决这个问题的一种方法是在一个文件中具有唯一 ID，但这样增加了存储空间并且使得管理稀少目标 ID 更困难。这个问题是通过允许每个场景使用它自己的目标 ID 解决的并且当来自一个场景的包指示跳到另一个场景时，它规定来自每个场景 ID 之间的目标映射。当从新的场景读出各个包时，该映射被用于转换该目标 ID。

期望目标映射信息将在与 JUMPTO 命令相同的包中。如果这个

信息是不可用的，则该命令被简单地忽略。目标映射可以由两个阵列表示：一个用于将在流中遇到的源目标 ID，和另一个用于目的目标 ID，这个 ID 是将由源目标 ID 变换的。在当前流中如果存在目标映射，则新的映射的目的 ID 利用当前流的目标映射阵列进行变换。如果在包中不规定目标映射，则新的流继承当前流（它可能是零）的目标映射。在一个流中的所有目标 ID 应当进行变换。例如，诸如基本首端 ID、其他 ID、按钮 ID、copyFrame（复制帧）ID、和重叠 ID 的各个参数应当都被变换为目的目标 ID。

在远端服务器情况，如图 15 所示，该服务器是与客户机远离的，这样数据 64 将发码流到客户机。媒体播放机客户机 20 被设计为解码从服务器 24 接收的各个包并发回用户操作 68 到服务器。在这种情况下，远端服务器 24 负责响应用户操作（如点击一个目标），并且修改正在发到客户机的包流 64。在这种情况下，每个场景含有单一复用流（由一个或多个目标组成）。

在这种情况下，根据客户机请求，服务器 24 通过复用多个目标数据流实时地组成各个场景，以构成一个单一复用包流 64（对于任何规定场景），该流被码流发到该客户机用于重放。这种结构允许根据用户的交互，使正在重放的媒体内容进行改变。例如，两个视频目标可以进行同时重放。当用户点击或选择一个时，其改变为不同的目标，同时另一个视频目标仍然不变。每个视频可以来自不同的源，使得服务器打开各个源和交叉比特流两者，增加适当的控制信息并传送新组成的流到客户机。服务器的责任是在码流发到客户机之前适当修改该流。

图 15 是远端码流服务器 24 的框图。如所示，类似于本地服务器，远端服务器 24 具有 2 个主要功能部分：数据流管理器 26 和动态媒体合成引擎 76。但是，智能复用器 27 可以例如从多个数据流管理器 26 提取输入，每个具有单一数据源并来自动态媒体组合引擎 76，而不是来自具有多输入的单一管理器。连同从各个源被复用在一起的目标数据包，智能复用器 27 插入附加各个控制包到包流中，以控制组成的场

景中的各个组成目标的修饰。远端数据流管理器 26 也是比较简单的, 仅执行顺序访问。此外, 远端服务器包括 XML 分析器 28, 能够通过 IAVML 源程序 (script) 29 可编程地控制动态媒体组合。远端服务器还从服务器操作员数据库 19 接收多个输入, 进一步控制和订制动态媒体组合处理 76。可能的输入包括: 每天的时间、每周的星期几、每年的天、客户机的地理位置, 和用户的人口统计数据, 诸如性别、年龄、任何存储的用户概况等。这些输入可以利用 IAVML 源程序作为变量以有条件地表达式而实现。远端服务器 24 还负责传送诸如目标选择之类的用户交互信息并形成数据返回到服务器操作员数据库 19, 用于以后接着的处理, 诸如数据小型化等等。

如图 15 所示, DMC 引擎 76 接收 3 个输入和提供 3 个输出。输入包括: 以 XML 为基础的源程序、用户输入和数据库信息。XML 源程序被用于通过规定如何组成正在码流发到客户机 20 的场景控制 DMC 引擎 76 的操作。组合是通过来自用户与在当前场景中的各目标的交互的可能而传递, 或来自分别的数据库的输入而传递的, 这些当前场景中的目标具有加在上面的 DMC 控制操作。这个数据库可以含有涉及日期时间、客户机的地理位置、或用户概况的信息。该源程序可以根据这些输入的任何组合控制动态组合处理。这是通过指令数据流管理器打开到 DMC 操作的连接和读出 DMC 操作所需的适当目标数据请求, 由 DMC 处理执行的, 它还指令智能复用器修改它的从数据流管理器接收的各目标包的交叉, 并且指令 DMC 引擎 76 实现一个场景中的目标的去除、插入或替代。DMC 引擎 76 还根据对每个在源程序中的目标控制规定, 选择性地产生和加控制信息到各目标上, 并且提供这个信息到智能复用器, 用于作为目标的一部分发码流到客户机 20。因此, 由 DMC 引擎 76 执行所有处理并且没有工作由客户机 20 执行, 而不是根据由任何目标控制信息提供的参数修饰自含目标。DMC 引擎 76 能够交替一个场景中的各个目标和视频中的各个场景两者。

与这种处理不同 MPEG4 要求执行相似功能。这种处理不利用源程序语言, 而依赖 BIFS。因此, 场景的任何修改都要求如下分别的修



改/插入操作：(i) BIFS、(ii) 目标描述符、(iii) 目标构形信息、和(iv) 视频目标数据包。BIFS 必须在客户机中利用特定的 BIFS-命令协议进行更新。因为 MPEG4 具有分别的而不互相依赖的数据部分定义一个场景，在组成成分中的改变不能通过简单地复用各目标数据包（有或没有控制信息）到一个包流实现，而要求远端操纵 BIFS、数据包的复用和构形信息、和新的目标描述符包的产生和传送。另外，如果对 MPEG4 目标要求先进的交互功能，分别写的 Java 程序被发送到 BIFS，用于由客户机执行，这样明显加重处理的额外开销。

如图 16 所示的流程图描述了本地客户机执行动态媒体组合 (DMC) 操作。在步骤 s301，客户机 DMC 处理开始并立即开始提供目标组合信息到数据流管理器、简化多目标视频重放，如步骤 s302 所示。DMC 检查用户命令表和其它各多媒体目标的可用性，保证视频仍然在播放（步骤 s303）；如果没有更多数据或用户已经停止视频重放，则客户机 DMC 处理结束（步骤 s309）。在步骤 s303，如果视频重放继续，则 DMC 处理将浏览启动 DMC 动作的用户命令表和目标控制数据。如步骤 s304 所示，如果没有被启动的动作，则处理返回步骤 s302 并且视频重放继续。但是，如果在步骤 s304 已经启动 DMC 动作，则 DMC 处理检查对象多媒体目标的位置，如步骤 s305 所示。如果对象目标本地存储，则本地服务器 ADMC 处理发送指令到本地数据源管理器，从本地源端出修改的目标流，如步骤 s306 所示，然后处理返回步骤 s304 检查进一步启动的 DMC 动作。如果对象目标存储在远端，则 MDC 处理发送适当的 DMC 指令到远端服务器，如步骤 s308 所示。另外一种方案，DMC 动作可以请求来源于本地和远端两者的对象目标，如步骤 s307 所示，因此适当的 DMC 动作由本地 DMC 处理执行（步骤 s306），并且 DMC 指令被发送到远端服务器来处理（步骤 s308）。从这种讨论中清楚看出，本地服务器支持混合、多目标视频重放，其中源数据是从本地和远端两者传送的。

利用如图 17 所示的流程图描述动态媒体组合引擎 76 的操作。在步骤 s401 开始 DMC 处理，并且在步骤 s402 进入等待状态，直至接

收到 DMC 请求。在接收到一个请求以后，在步骤 s403、s404 和 s405，DMC 引擎 76 查询请求类型。如果在步骤 s403 该请求被确定为目标替换动作，则存在两个对象目标，一个动作对象目标和一个新的对象目标将被加到流中。首先，在步骤 s406 指令数据流管理器从复用的比特流中删除动作对象目标包，并停止从存储器中读动作对象目标流。其次，在步骤 s408 指令数据流管理器从存储器中读新的对象目标流，并且交叉这些包到发送的复用比特流中。然后，在步骤 s402，DMC 引擎 76 返回其等待状态。如果在步骤 s403 请求不是目标替换动作，在步骤 s404 如果动作类型是目标去除动作，则是动作对象目标的一个对象目标存在。在步骤 s407 处理目标去除动作，在该步骤指令数据流管理器从复用比特流中删除动作对象目标包，并停止从存储器中读动作对象目标流。然后，在步骤 s402，DMC 引擎 76 返回等待状态。在步骤 s404，如果请求的动作不是目标去除动作，在步骤 s405 如果动作是目标加入动作，则是新的对象目标的一个对象目标存在。在步骤 408 处理目标加入动作，在该步骤指令数据流管理器从存储器读新的对象目标流，并且交叉这些包到发送的复用比特流中。然后，在步骤 s402，DMC 引擎 76 返回等待状态。最后，请求的 DMC 动作不是目标替换动作（在步骤 s403）、或不是目标去除动作（在步骤 s404）、或不是目标加入动作（在步骤 s405），则在步骤 s402，DMC 引擎 76 忽略请求并返回等待状态。

## 视频解码器

存储、发送和处理未加工的视频数据是低效率的，这样计算机视频系统通常编码视频数据为压缩格式。下面这部分是描述如何将视频数据编码为有效的压缩形式。这部分描述负责从压缩的视频数据流中产生视频数据的视频解码器。该视频编解码器支持任意构形的视频目标。它利用 3 个信息分量表示每个视频帧：一个色图、一个树形编码位映像、和一个运动矢量表。该色图是在帧中使用的所有的色的表，该表是按分配给红、绿和兰每个分量的 8 比特的具有 24 比特的精度。

通过它们的索引这些色被引入到色图中。位映像被用于定义许多部分，包括在显示时将被修饰的帧中的各个像素的色、将被作成透明的帧的区域、和将不改变的帧的区域。在编码帧中的每个像素可以分配给这些功能之一。通过它的值定义一个像素起到这些功能中的哪个的作用。例如，如果使用 8 比特色表示，则色值 0xFF 可以被分配给指示对应于场景上的像素，它从其当前值是不改变的，并且色值 0xFE 可以被分配给指示对应于场景上的该目标将是透明的像素。编码的帧像素色值指示它是透明的一个场景的情况下，屏上像素的最后色取决于背景场景色和任何在下面的视频目标。下面描述对于这些分量的每个使用的构成编码视频帧的具体编码。

通过首先发送一个整数值到比特流以指示接着的表项目号来编码色表。将要发送的每个表项目然后由首先发送的其索引进行编码。接着，对每个色分量（Rf、Gf 和 Bf）发送 1 比特标志，指示：如果该标志是 ON（通），则色分量被全字节发送，并且如果该标志是 OFF（断），则将发送代表色分量的高序半字节（4 比特）和低序半字节设置为 0。因此，按照下列图形对表项目进行编码，其中括号中的数或 C 语言指示发送的比特数 R（Rf? 8: 4）、G（Gf? 8: 4）、B（Bf? 8: 4）。

运动矢量按照一个阵列进行编码。首先，发送按 16 位值的阵列中的各运动矢量的号，接着是宏块的尺寸，然后是运动矢量的阵列。阵列中的每个项目含有宏块位置和块的运动矢量。运动矢量按 2 个带符号的矢量的每个水平和垂直分量之一的半字节进行编码。

利用预定树遍历方法编码实际视频帧数据。在树的叶上存在两个类型：透明叶和区域色叶。透明叶指示由叶表示的屏上显示区是不变的，而色叶将强迫屏上区为由叶规定的色。根据这 3 个功能，可以按照事先的描述分配任何编码的像素，透明叶将对应于 0xFF 的色值，而表示屏上区被强迫为透明的具有 0xFE 值的像素被处理为正常区色的叶。该编码器在树的顶端开始并在每个节点存储单一比特，指示是否该节点是一个叶或一个父体。如果是叶，这个比特值设置为 ON

(通), 并且发送另外的单一比特, 指示是否该区是透明的 (OFF), 否则接着另一个比特标志设置为 ON, 指示是否该叶的色作为一个索引被发送到 FIFO 缓冲器或者作为实际索引发送到色图。如果这个标志被设置为 OFF, 则两个比特的码字作为 FIFO 缓冲器的各项目之一的索引被发送。如果标志 ON, 则这指示在 FIFO 中未找到叶色, 并且实际色值被发送并且还插入到 FIFO, 推出存在的各项目之一。如果该树节点是父节点, 则存储一个单一 OFF 比特, 并且然后利用相同方法, 4 个子节点的每个单个地进行存储。当编码器到达树的最低级别时, 则所有节点是叶节点并且不使用叶/父体指示比特, 代之以存储第一叶色码字接着的透明比特。比特图形的发送可以被表示为如下。使用如下的各个符号: 节点类型 (N)、透明 (T)、FIFO 预测色 (P)、色值 (C)、FIFP 索引 (F)

$$\begin{aligned}
 &N(1) \dots \text{off} \rightarrow N(1) [\dots], N(1) [\dots], N(1) [\dots], N(1) [\dots] \\
 &\quad \backslash \dots \text{on} \rightarrow T(1) \dots \text{off} \\
 &\quad \quad \backslash \dots \text{on} \rightarrow P(1) \dots \text{off} \rightarrow F(2) \\
 &\quad \quad \quad \backslash \dots \text{on} \rightarrow C(x)
 \end{aligned}$$

图 49 是表示视频帧解码处理一个实施例的之一步骤的流程图。对压缩比特流的视频帧解码处理在步骤 s2201 开始。在步骤 s2202 从比特流中读出压缩比特流中用于物理地分开各种信息分量的层识别符。如果层识别符指示运动矢量数据层开始, 则步骤 s2203 前进到 s2204, 从比特流中读出和解码运动矢量, 并且执行移动补偿。运动矢量被用于从以前缓冲的帧中将指示的宏块复制到叶该矢量指示的新位置。当移动补偿处理完成时, 在步骤 s2202 从比特流读出下一层指示符。如果该层指示符指示 4 元树数据层开始, 步骤 s2205 前进到步骤 s2206, 并且利用读出的叶色处理启动 FIFO 缓冲器。接下来, 在步骤 s2207, 从压缩比特流中读出 4 元树的深度, 并用于启动该 4 元树的 4 个侧面。在步骤 S2208, 压缩的位映像 4 元树现在被解码。由于 4 元树的数据

被解码, 根据叶值修改帧中的区域值。它们可能被用新的色重写、设置透明、或保持不变。当 4 元树数据被解码时, 在步骤 s2202, 解码处理从压缩比特流中读下一层识别符。如果该层指示色图数据层开始, 步骤 s2209 前进到步骤 s2210, 从压缩比特流中读出将被更新的色数。在步骤 s2211 如果存在一个或多个色更新, 则在步骤 s2212 从压缩比特流中读出第一色图索引值, 并在步骤 s2213 从压缩比特流中读出色分量值。每个色更新又通过步骤 s2211、s2212、和 s2213 读出, 直至所有色更新均已被执行, 此时步骤 s2211 前进到步骤 s2202, 从压缩比特流中读出新的层识别符。如果层识别符是一个数据结束识别符, 步骤 s2214 前进到步骤 s2215 并结束视频帧解码处理。如果通过步骤 s2203、s2205、s2209 和 s2214 层识别符是未知的, 则层识别符被忽略, 并且处理返回 s2202 读下一个层识别符。

图 50 是表示具有底级别节点类型消除的 4 元树解码器主要步骤的流程图。这个流程图实现一种递归方法, 对于处理的每个树的象限递归地调用自身。4 元树解码方法从步骤 s2301 开始, 具有与识别将被解码的象限的深度和位置相同的机制。在步骤 s2302 中, 如果该象限是非底部象限, 则在步骤 s2307, 从压缩比特流中读节点的类型。在步骤 s2308, 如果节点类型是父节点, 则然后在步骤 s2309 对顶部左象限、在步骤 s2310 对顶部右象限、在步骤 s2311 对底部左象限、在步骤 s2312 对底部右象限, 4 个递归调用作出 4 元树解码处理, 接着在步骤 s2317 该解码处理的迭代结束。对每个象限的作出的递归调用的具体次序是任意的, 但是, 该次序与由编码器执行的 4 元树去组合处理的次序是相同的。如果该节点类型是一个叶节点, 处理从步骤 s2308 到 s2313 继续, 并从压缩比特流中读出叶类型值。在步骤 s2314, 如果该叶类型值指示一个透明叶, 则在步骤 s2317, 解码处理结束。如果该叶是不透明的, 则在步骤 s2315 从压缩比特流中读出叶色。在本说明书中, 叶读出色值功能使用 FIFO 缓冲器。接下来, 在步骤 s2316, 该图象树被设置适当叶色值, 这可能是背景目标色或按指示的叶色。在图象更新完成以后, 在步骤 s2317, 这次迭代 4 元树解码功

能结束。递归调用 4 元树解码功能继续，直至达到底层象限。在这一层不存在需要包括压缩比特流中的父体/叶节点的指示符，因为在这一层的每个节点都是叶，因此步骤 s2302 前进到步骤 s2303，并立即读出叶类型值。如果在步骤 s2304 叶是不透明的，则在步骤 s2305 从压缩比特流中读出叶色值，并且在步骤 s2306 图象象限色被适当更新。这种解码处理的迭代在步骤 s2317 结束。4 元树解码处理的递归处理实行继续，直至压缩比特流中的所有叶节点都已被解码。

图 51 表示读 4 元树叶色执行的步骤，在步骤 s2401 开始。在步骤 s2402 从压缩比特流中读单一标志。这个标志指示该叶色是将从 FIFO 缓冲器还是直接从比特流读出的。在步骤 s2403 如果叶色不是从 FIFO 读出的，则叶色是在步骤 s2404 从压缩比特流读出的，并且在步骤 s2405 被存储在 FIFO 缓冲器中。在 FIFO 中存储的新读出的色推出 FIFO 中最近加入的色。在更新 FIFO 后于步骤 s2408，读取叶色功能结束。然而，如果该叶的色已经被存储在 FIFO 中，则在步骤 s2406 从压缩比特流读出 FIFO 索引码字。在步骤 s2407，根据当前读出的码字通过检索 FIFO 确定该叶色。在步骤 s2408 结束读出叶色的处理。

## 视频编码器

到此为止，讨论了针对预存在的视频目标和含有视频数据的文件的操作。以前的部分，描述了如何将压缩的视频数据进行解码产生未加工的视频数据。在这部分中，讨论产生这种数据的处理。该系统被设计为支持多个不同编解码器。这里描述两个这样的编解码器，其它方案还可以用于包括 MPEG 族和 H.261 和 H.263 及其后继版本。

该编码器包括如图 18 所示的各个主要部分。这些部分是按软件实现的，但为了增加编码器的速度，所有部件可以用特别开发执行编码处理的各个步骤的专用集成电路 (ASIC) 实现。音频编码部分 12 压缩输入音频数据。根据 ITU 规范 G.723 或者 IMA ADPCM 编解码器，音频编码部分 12 可以使用自适应增量脉冲编码调制 (ADPCM)。场景/目标控制数据部分 14 编码与输入音频和视频相关的场景动画和表

示参数，确定每个视频目标的关系和行为。输入色处理部分 10 接收和处理单个的输入视频帧并消除残余和不希望的色。这种处理还从图象中去掉不希望的噪声。可以选择地，利用以前编码帧作为基础，对输入色处理器 10 的输出执行移动补偿。色差管理和同步部分 16 接收输入色处理器 10 的输出，并根据以前编码帧作为基础确定利用任选移动补偿进行编码。然后，输出被提供到组合空间/时间编码器 18 以压缩视频数据，并被提供到执行反向功能的解码器 20 用以在一个帧延迟 24 之后提供帧到移动补偿部分 11 两者。发送缓冲器 22 接收空间/时间编码器 18、音频编码部分 12 和控制数据部分 14 的输出。发送缓冲器 22 经速率信息反馈到组合空间/时间编码器 18，通过交叉编码数据和控制数据速率，管理来自装在编码器中的视频服务器的发送。如果需要，编码数据可以由用于传输的加密部分 28 进行加密。

图 19 的流程图描述编码器执行的主要步骤。视频压缩处理在 s501 开始，输入帧压缩环路（s502 到 s521），并当在步骤 s502 输入视频数据流中没有剩余视频数据帧时，在步骤 s522 结束。在步骤 s503 取出来自输入数据流的未加工视频帧。此时，可能希望执行空间过滤。执行空间过滤以降低正在产生的视频的比特率或总位数。但是空间过滤还降低保真度。如果在步骤 s504 确定执行空间过滤，则在步骤 s505 计算当前输入视频帧与以前处理过的或重建过的视频帧之间的色差帧。最好是执行其中存在着运动的空间过滤，以及计算帧差的步骤指示存在着运动的，如果没有差，则不存在运动，并且一个帧在各区域中的差指示这些区域的运动。接下来，在步骤 s506，对输入视频帧执行局部空间过滤。这种过滤是局部的，使得仅各帧之间有改变的图象区被过滤。如果希望，也可以对 I 帧执行空间过滤。这可以利用任何希望的技术进行，包括：例如反向梯度过滤、中间过滤、和/或这两种类型的组合过滤。在步骤 s505 中，如果希望对一个关键帧执行空间过滤并且还计算帧差，则用于计算帧差的参考帧可以是一个空帧。

在步骤 s507 执行色量化，从图象去除统计上不重要的色。对于静止图象的色量化的一般处理是公知的。可以使用在本发明中的色量化

的举例类型的所有技术包括但不限于描述在和援引于美国专利 5432893 和 4654720 的所有技术中，这些专利援引于此以资参考。还包含供参考的是在这些专利中援引和参考的所有文献。关于步骤 s507 的色量化的进一步的信息参考图 20 的单元 10a、10b 和 10c 进行说明。如果对这一帧执行色图更新，流程从步骤 s508 前进到步骤 s509。为了实现高质量的图象，色图可以每帧被更新。但是，这样可能导致太多信息要被传输，或者可能要求太多的处理。因此，替代更新每帧的色图，色图以每  $n$  帧被更新，其中  $n$  是等于或大于 2 的整数，最好是小于 100，并更好是小于 20。另一种方案是，色图可以被平均每  $n$  帧更新，其中  $n$  不要求是一个整数，而可以是包括大于 1 和小于诸如 100 和更好是小于 20 的预定数的分数的任何值。这些数仅是示例性的并且，如果希望，按照需要色图可以经常或不经常地更新。

当希望更新色图时，在步骤 s509 执行新色图的选择和与以前帧的色图进行相关。当色图改变或被更新时，则希望保持当前帧的色图类似于以前帧的色图，以便没有使用不同色图的各个帧之间的视觉上的不连续性。

如果在步骤 s508 没有色图（例如，不需要更新色图），则以前帧的色图被选择或使用于这一帧。在步骤 s510，根据选择的色图，量化的输入图象色被重新映射为新的色。步骤 s510 对应于图 20 的方框 10d。接下来，在步骤 s511 执行帧缓冲器交换。步骤 s511 的帧缓冲器交换促进更快和更高存储器效率的编码。作为帧缓冲器交换的实施例子，可以使用两个帧缓冲器。当一个帧已经被处理时，用于这个帧的缓冲器被指定保持过去的帧，和一个在另外的缓冲器接收的新帧被指定为当前帧。这种帧缓冲器的交换允许有效的存储器分配。

一个关键参考帧，也称为参考帧或关键帧可以用作参考。如果步骤 s512 确定这个帧（当前帧）将被编码为，或者指定为关键帧，则视频压缩处理直接到步骤 s519，以编码和发送该帧。一个视频帧可以被编码为关键帧，由于如下原因，包括：（i）接着视频定义包的视频帧序列中的第一帧，（ii）该编码器检测视觉场景在视频内容的变化，



或 (iii) 用户已经选择将插入到视频包流中的关键帧。如果该帧不是关键帧, 在步骤 s513 视频压缩处理计算当前色图索引的帧与以前重建色图索引的帧之间的差帧。该差帧、以前重建的色图的索引的帧、和当前色图索引的帧被使用在步骤 s514 中, 以产生运动矢量, 在步骤 s515 该运动矢量又被使用于重排以前帧。

在步骤 s516 比较重排的以前帧和当前帧, 产生有条件的补充图象。如果在步骤 s517 能实现兰屏透明, 则 s518 将脱离落入兰屏阈值的差帧的各区域。在步骤 s519, 现在该差帧被编码和发送。步骤 s519 下面参照图 24 将进一步说明。在步骤 s520 根据编码比特流的尺寸建立比特速率控制参数。最后在步骤 s521 重建编码帧, 用于在步骤 s502 开始编码下一个视频帧。

图 18 的输入色处理部分 10 执行统计上的不重要色的减少。选择执行这种色减少的色空间是不重要的, 因为利用多个不同色空间的任何一个可以实现相同的结果。

如上所述, 利用各种矢量量化技术可以实现统计上不重要色的减少, 并且利用包括描述在作者为 S.J.Wan, P.Prusinkiewicz, S.KM.Wong 的出版物 “Variance-Based Color Image Quantization for Frame Buffer Display”, (*Color Research and Application*, Vol.15, No.1, Feb 1990) 中的普及的、中央分割的、k-最近似邻近值的和方差方法的任何其它技术也可以实现, 该出版物援引于此, 以资参考。如图 20 所示, 这些方法可以利用一种初始统一或非自适应量化步骤 10a 通过减少矢量空间, 改善矢量量化算法 10b 的性能。如果需要, 方法的选择是保持各个量化视频帧之间的最高时间相关量。到这种处理的输入是候选视频帧, 并且通过分析个帧中的统计色分配使该处理继续进行。在 10c, 选择用于代表图象的各个色。利用某些手工处理装置或者个人数字帮助的现在可用的技术, 同时显示例如 256 色可能存在着限制。因此, 10c 可以用于选择将用于代表图象的 256 个不同色。矢量量化处理的输出是一个表示整个帧 10c 的各个色的表, 该帧可以在尺寸上进行限制。在普及方法的情况下, 选择最经常 N 个色。

最后，原始帧中的每个色被重新映射 10d 到表示集中的各个色之一。

输入色处理部分 10 的色管理部分 10b、10c 和 10d 管理视频中的色变化。输入色处理部分 10 产生含有一组显示色的表。这组色随时间动态地变化，规定处理是以每帧为基础自适应的。这允许各个视频帧的色成分不降低图象质量地改变。选择适当的方案管理色图的自适应是重要的。对于每个色图存在 3 个不同的可能性：可以是静态的、分段和部分静态的、或全动态的。利用固定或静态的色图，局部色质量将被降低，但保存帧间的高相关性，导致高的压缩增益。为了保持在视频场景可能经常变化情况中的高质量图象，色图应当能够立即自适应。对每帧任选新的最佳色图具有高的带宽要求，因为不仅色图要每帧更新，而且还有大量的像素每次应将需要重新映射。这种重新映射还引入色图闪烁的问题。一种折中是仅允许在连续帧之间的有限的色变化。这可以通过分割色图为静态和动态部分，或者通过限制每帧允许变化的色数来实现。在第一种情况下，可以修改表的动态部分的项目，它保证某些预定义的色始终是可用的。在另外一种方案中，不保留各个色和可修改任何色。同时这种方法帮助保存某些数据相关性，色图可能不能迅速地足以在某些情况下消除图象质量恶化的自适应。现存的各个方法折中图象质量与保存帧间的图象相关性。

对于任何这些动态色图的方案，对保存时间上的相关性同步是重要的。这种同步处理具有 3 个部分：

1. 保证随着时间从一帧转到下一帧的色被映射到相同的索引。这涉及采用与当前色图相关的每个新色。
2. 一种替换方案被用于更新改变的色图，降低色闪烁的量，最适合的方案是利用最相似的新替代色替代绝对色。
3. 最后，在图象中对不再受到支持的任何色的所有现存的参考由对当前支持色的参考替代。

接着图 18 的输入色处理 10，视频编码器的下一个部分索引的各个色帧并任选地执行运动补偿 11。如果不执行运动补偿，则来自帧缓冲器的 24 的以前帧不由运动补偿部分 11 修改并直接传送到色差管理

和同步部分 16。优选的运动补偿方法由分段视频帧为小块开始，并确定视频帧中像素数需要补充或更新和超过某个阈值是不透明的所有块。然后，对产生的像素块执行运动补偿处理。首先，在该区的邻近进行搜索，确定是否该区从以前帧已经被取代。对于执行这种操作的传统方法是计算参考区与候选替换区之间的均方误差（MSE）或和误差平方和（SSE）度量。如图 22 所示，可用利用一种穷举搜索或多个其它现存搜索技术之一执行这种处理，这些技术诸如 2D 对数 11a、3 步骤 11b、或简化的共扼方向搜索 11c。这种搜索的目的是找到对该区的替代矢量，通常叫做运动矢量。传统的度量不利用索引的/色映射图象表示，因为这些方法依靠提供连续图象表示的连续性和空间-时间相关性。利用索引表示，存在非常小的空间相关并且从一帧到一帧非逐渐地或连续地像素色改变，相反，改变是不连续地，按照色索引跳到新的色图项目，反映像素色的变化。因此，单索引/像素变化色将引入对于 MSE 或 SSE 的大的变化，降低了这些度量的可靠性。因此，对于定位区替换的较好的度量是如果该区不透明以前帧与当前帧区比较的差最小的像素数。找到一旦运动矢量，根据运动矢量，通过从其在以前帧中的原始位置预测该区中的像素值。如果给出最小差的该矢量对应于非替换，该运动矢量可能是 0。对于每个替换块的运动矢量，连同其相关块地址被编码到输出比特流。接着，色差管理器部分 16 计算运动补偿以前帧与当前帧之间的感觉差。

色差管理器部分 16 负责计算当前的与以前帧之间的每个像素的感觉色差。这种感觉色差是根据对感觉色降低所描述的相似方法计算的。如果它们的色已经变化超过规定量，则更新像素。色差管理器部分 16 还负责清除图象中所有无效色图参考，并利用有效参考替换这些，产生一个有条件的补充图象。当在色图中较新的色替代老的色时，可能发生无效色图参考。然后，这种信息被传到视频编码处理中的空间/时间编码部分 18。这种信息指示在帧中哪个区是全透明的，并且哪个需要替换，并且在色图中哪个色需要更新。帧中不更新的所有区由设置像素为已经选择表示不更新的预定值予以识别。这个值的含义允

许任意构形的视频目标的产生。为了保证预测误差不累积并不恶化图象质量，使用环路过滤器。这强迫帧补充数据将从该出现的帧和累积的以前发送数据（解码图象的当前状态）来确定，而不是从出现的和以前的帧确定。图 21 提供色差管理部分 16 的更详细的图。当前帧存储器 16a 含有从输入色处理部分 10 产生的图象。以前帧存储部分 16b 含有由 1 帧延迟部分 14 缓冲的帧，它可以有或者可以没有由运动补偿部分 11 进行的运动补偿。色差管理部分 16 分为 2 个主要部分：各个像素之间的感觉色差的计算 16c，和无效色图参考的清除 16f。感觉色差相对于阈值 16d 进行评估，确定哪些像素需要更新，并且产生的像素被任选地过滤 16e，以降低数据速率。从过滤空间过滤器 16e 和无效色图参考 16f 的输出形成最后更新的图象 16g，并且被发送到空间编码器 18。

这产生现在编码有条件补充帧。空间编码器 18 利用树分裂方法，根据分裂的标准，递归地分割每个帧为较小的多边形。如图 23 所示使用 4 元树分裂 23d。在一个第 0 序插值的例子中，这试图通过统一块表示图象 23a，其值等于该图象的整体平均值。在另外一个例子中，可以使用第一和第二序插值。在图象的某个位置上，如果这个表示值与实际值之间的差超过某个容限的阈值，则该块被递归地均匀地细分为 2 个或 4 个子区，并且对每个子区计算新的平均值。对于无损失地图象编码，不存在容限阈值。树结构 23d、23e、23f 是由各个节点和各个指针组成的。其中每个节点表示一个区并含有对任何可能存在的代表子区的子节点的各个指针。有 2 个类型的节点：叶 23b 和非叶 23c 节点。叶节点 23b 是不进一步分解并从而没有子女的，代之以含有暗示区的值的这样的节点。非叶节点 23c 不含有代表值，因为这些节点叶进一步由各个子区组成并且从而含有相应各个子节点的指针。这些节点也可以叫做父节点。

### 动态位映像（色）编码

单一视频帧的实际编码表示包括：位映像、色图、运动矢量和视

频增量数据。如图 24 所示，视频帧编码处理从步骤 s601 开始。如果 (s602) 运动矢量经由运动补偿处理产生，则在步骤 s603 对运动矢量编码。如果因为以前的视频帧 (s604) 色图已改变，则在步骤 s605 编码新的色图项目。在步骤 s606 从位映像帧产生树结构并在步骤 s607 进行编码。如果 (s608) 视频增量数据将进行编码，则在步骤 s609 编码增量数据。最后，在步骤 s610 结束视频帧编码处理。

实际 4 元树视频帧数据利用预排序树遍历方法 (preordered tree traversal method)。树中可能有 2 种类型的叶：透明叶和区域有色叶。透明叶表示由叶指示的区从以前值不改变（这些不存在于视频关键帧中），并且色叶含有区色。图 26 表示预定树遍历编码方法，用于具有第 0 序插值和底层节点类型消除的正常预测的视频帧。图 26 的编码器在步骤 s801 开始，在步骤 s802 初始地加 4 元树层识别符到编码的比特流上。在树的顶部开始，步骤 s803，编码器得到初始节点。在步骤 s804，如果节点的父节点，则在步骤 s805，编码器加一个父节点指示符（一个单 0 比特）到比特流中。接着，在步骤 s806，从树取下一个节点，并且编码处理返回步骤 s804，编码树中的接下来的各个节点。在步骤 s804，如果节点不是父节点，即是叶节点，则在步骤 s807 编码器检查该节点在树中的级别。在步骤 s807，如果该节点不是树的底部，则在步骤 s808，编码器即加一个叶节点标志（一个单一的 1 比特）到比特流中。在步骤 s809，如果该叶区是透明的，则在步骤 s810，透明叶标志（一个单 0 比特）被加到比特流中。否则，在步骤 s811，一个不透明叶标志（单 1 比特）被加到比特流中。然后在步骤 s812，不透明叶色被按如图 27 所示编码。但是，在步骤 s807，如果叶节点是在树的底部级别，则发生底部级别节点类型的消除，因为所有节点都是叶节点，并且没有使用叶/父体指示比特，使得在步骤 s813，4 个标志都被加到比特流中，以指示在这个级别上的 4 个叶的每一个是透明的 (0) 或是不透明的 (1)。接下来，在步骤 s814，如果顶部的左叶是不透明的，则在步骤 s815 顶部的左叶的色被按如图 27 所示编码。在这个第二底部级别，每个叶节点重复步骤 s814 和 s815 的每一步，对

于顶部右节点，如在步骤 s816 和 s817 所示，对于底部左节点，如在步骤 s818 和 s819 所示，对于底部右节点，如在步骤 s820 和 s821 所示。在叶节点被编码以后（从步骤 s810、s812、s820 或 s821），在步骤 s822，编码器检查在树上是否还有剩余的节点。如果在树上没有剩余的节点，则在步骤 s823 编码处理结束。否则，在步骤 s806 编码处理继续，在该步骤从树上选择下一个节点，并且从步骤 s804 对新的节点整个处理重新开始。

在视频关键帧（没有预测）的特殊情况下，如图 28 所示，这些不具有透明的叶和使用稍不同的编码方法。关键帧编码处理从步骤 s1001 开始，在步骤 s1002 初始地加 4 元树级别指示符到编码的比特流中。在树的顶部开始，步骤 s1003，编码器得到初始节点。在步骤 s1004，如果该节点是父节点，则步骤 s1005 编码器加父节点标志（单一 0 比特）到比特流中，接下来，在步骤 s1006 从树取下一个节点，并返回步骤 s1004 的编码处理，以编码树的下一个节点。但是，如果在步骤 s1004 该节点不是父节点，即是叶节点，则在步骤 s1007 编码器检查该节点在树上的节点级别。在步骤 s1007 如果该节点大于树的从底部一个级别，则在步骤 s1008 编码器加一个叶节点标志（一个单一 1 比特）到比特流。然后，在步骤 s1009 按如图 27 所示编码不透明叶色。但是，在步骤 s1007 如果该叶节点是从树的底部的一个级别，则出现底部级别节点类型消除，因为所有节点都是叶节点并且不使用叶父体指示比特。因此，在步骤 s1010，顶部左叶色被按如图 27 所示编码。接下来，在步骤 s1011、s1012 和 s1013，不透明叶色被分别类似于顶部右叶、底部的左叶和底部的右叶编码。在叶节点被编码（从步骤 s1009 或 s1013）以后，在步骤 s1014 编码器检查是否还有节点在树上。如果没有节点在树上，在步骤 s1015 编码处理结束。否则，在步骤 s1016 编码处理继续，在该步骤从树选择下一个节点，并且对新的节点从步骤 s1004 整个处理重新开始。

利用 FIFO 缓冲器如图 27 所示对不透明叶色进行编码。叶色编码处理从步骤 S901 开始。要编码的色与已经在 FIFO 中的 4 个色比较。

在步骤 s902 如果确定该色在 FIFO 缓冲器中,则在步骤 s903 单一 FIFO 查找标志 (单一 1 比特) 被加到比特流中,接着在步骤 s904 作为索引表示叶色的比特码字进入 FIFO 缓冲器。这个码字索引 FIFO 缓冲器中 4 个项目之一。例如, 00、01 和 10 的索引值分别规定该叶色与以前叶、此前的以前不同叶色、和此前的以前叶色相同。但是,在步骤 s902 如果待编码的色是 FIFO 中不可用的,则在步骤 s905 发送的色标志 (单一 0 比特) 被加到比特流中,在步骤 s906 接着发送表示顶部实际色值的 N 比特。另外,该色被加到 FIFO,推出现存的各个项目之一。然后,色叶编码处理在步骤 s907 结束。

色图被类似地压缩。标准表示是按 24 比特发送每个索引,其中 8 比特规定红分量值、8 比特用于绿分量和 8 比特用于兰分量。在压缩格式中,单一比特标志指示是否每个色分量是按照全 8 比特值规定的,或者刚好顶部半字节与底部 4 比特设置为 0。接着这个标志,分量值按照 8 或 4 比特根据该标志进行发送。图 25 的流程图描述利用 8 比特色图索引的色图编码方法的一个实施例。在这个实施例中,在各个色分量之前,规定一个色的所有分量的色分量分辨率的单一比特标志被编码。色图更新处理在步骤 s701 开始。开始,在步骤 s702 色图层识别符被加到比特流中,在步骤 s703 接着发送指示更新的色数。在步骤 s704 该处理检查附加更新的色更新表,如果没有另外的色更新要求编码,则处理在步骤 s717 结束。但是,如果仍有色要编码,则在步骤 s705 待更新的色表索引被加到比特流中。对于每个色一般存在多个分量(例如,红、绿和兰),因此步骤 s706 形成围绕步骤 s707、s708、s709 和 s710 的一个环路条件,分别处理每个分量。在步骤 s707 从数据缓冲器中读出每个分量。接下来,在步骤 s708 如果该分量的低序半字节是 0,则在步骤 s709,一个 off 标志 (单一 0 比特) 被加到比特流中,或如果低序半字节不是 0,则在步骤 s710,一个 on 标志 (单一 1 比特) 被加到比特流中。处理通过返回步骤 s706 重复,直至没有剩余色分量。接下来,在步骤 s711 从数据缓冲器再次输出第一分量。同样,步骤 s712,形成围绕步骤 s713、s714、s715 和 s716 的环路条件,分别处理

每个分量。接下来，在步骤 s712 如果分量的低序半字节是 0，则在步骤 s713 分量的高序半字节被加到比特流。另一种方案，如果低序半字节不是 0，则步骤 s714 分量的 8 位色分量被加到比特流。在步骤 s715 如果还有色分量仍要相加，则在步骤 s716 从输入数据流读出下一个色分量，并且处理返回步骤 s712 处理这个分量。否则，在步骤 s715 如果没有剩余分量，则色图编码处理返回步骤 s704 处理任何剩余色图更新。

### 备用编码方法

在该备用编码方法中，处理非常类似于如图 29 所示的第一种方法，除了图 18 的输入色处理部分 10 不执行色减少，如果需要的话，而代之以保证输入色空间是按从 RGB 变换的 YCbCr 格式。没有要求的色量化或色图管理，因此图 19 的步骤 s507 到 s510 被单一的色空间变换步骤取代，保证帧是按 YCbCr 色空间表示的。图 18 的运动补偿部分 11 对 Y 分量执行“传统”运动补偿并存储运动矢量。然后从利用来自 Y 分量的各运动矢量对 Y、Cb、Cr 分量每个的帧间编码处理，产生有条件的补充图象。然后，通过在每个方向的 2 个系数下取样 Cb 和 Cr 位映像后，独立压缩 3 个结果的不同图象。位映像的编码使用一种类似的递归树分解，但这次对不在树底部的每个叶，存储 3 个值，平均对于由该叶代表的区的位映像值，和水平和垂直方向的梯度。图 28 的流程图描述备用位映像编码处理，从步骤 s1101 开始。在步骤 s1102 选择编码的图象分量（Y、Cb 或 Cr），然后在 s1103 选择初始树节点。在步骤 s1104，如果这个节点是父节点，则在步骤 s1105 将父节点标志（1 比特）加到比特流中。然后在步骤 s1106 从树选择下一个节点，并且备用位映像编码处理返回 s1104。在步骤 s1104 如果新的节点不是父节点，则在步骤 s1107 确定树的节点深度。在步骤 s1107 该节点不是树的底部级别的节点，则利用非底部叶节点编码方法编码该节点，使得在步骤 s1108 将叶节点标志（1 比特）加到比特流。接下来，在步骤 s1109 如果叶是透明的，则透明叶标志（1 比特）



被加到比特流。然而，如果叶是不透明的，则不透明叶标志（1比特）被加到比特流中，接下来在步骤 S1112 编码叶色平均值。与第一种方法一样，通过发送一个标志和按 2 比特的 FIFO 索引或者按 8 比特的平均值本身，该平均值利用 FIFO 编码。在步骤 s1113 如果该区不是不可见背景区（用于任意构形视频目标），则在步骤 s1114 编码叶水平和垂直梯度。利用平均值的特殊值，例如 0xFF 编码不可见的背景区。按照 4 位量化值发送该梯度。但是，如果在 S1117 确定该叶节点是在树的最底部级别，则通过发送位映像值和非父体/引导指示标志按以前的方法编码对应的叶。与以前一样利用单一比特标志编码透明叶和色叶。在任意构形视频的情况下，利用平均值的特殊值，例如 0xFF，编码不可见的背景区，并且在这种情况下，不发送梯度值。具体地，然后在步骤 s1154 个标志被加到比特流上，以指示是否在这个级别上的 4 个叶的每个是透明或者不透明的。接下来，在步骤 s1116，如果顶部叶是不透明的，则在步骤 s1117 该顶部左叶色按照上述不透明叶色编码进行编码。对这个底部级别的每个叶节点重复步骤 s1116 和 s1117 的每个，如所示对顶部右节点重复步骤 s1118 和 s1119 的每个所示，对底部左节点重复步骤 s1120 和 s1121 的每个，对底部右节点重复步骤 s1122 和 s1123 的每个。完成叶节点编码时，在步骤 s1124 编码处理检查树的另外节点，如果没有剩余节点编码则在步骤 s1125 结束。另外一种情况下，在步骤 S1106 取下一个节点，并且处理在步骤 S1104 重新开始。在这种情况下，重建包括利用第一、第二或第三序插值在由叶识别的每个区插入各个值并且然后组合对于 Y、Cb 和 Cr 分量每个的值，产生每个像素的 24 位 RGB 值。为了利用 8 位装置，映射色的显示、色的量化是在显示前执行的。

### 色预量化数据的编码

为了改善图象的质量，与在前面描述的备用编码方法一样，可以使用第一或第二序插值编码。在这种情况下，不仅对存储的叶每个叶代表区的是平均色，而且还是每个叶的色梯度信息。然后利用二次方

程或三次方程插值产生连续色调图象执行重建。当在装置利用索引的色显示连续显示色图象时，这可能产生一个问题。在这些情况下，需要下降到 8 位的量化输出和按实时地索引是禁止的。如图 47 所示，在这种情况下编码器 50 可以执行 24 位色数据 02a 的矢量量化 02b，产生色预量化数据。如下所述，利用 8 元树压缩 2c 可以编码色量化信息。这个压缩的色预量化数据随着编码的连续色调图象进行发送，通过施加预计算的色量化数据使得视频解码器/播放机 38 执行实时色量化 02d，因此产生可选 8 位索引的色视频实时表示 02e。当使用重建过滤产生将可以显示在 8 位装置上的 24 位结果时，还可以使用这种技术。这个问题通过发送少量信息到视频解码器 38 可以解决，该信息描述从 24 位色结果到 8 位色表的映射。这种处理描述在从步骤 s1201 开始的图 30 的流程图中，并且包括包含在客户机中执行实时色量化的预量化处理中的主要步骤。在步骤 s1202 视频中的所有帧按照通过条件块的指示被连续地处理。如果没有剩余帧，则在步骤 s1210 预量化结束。否则在步骤 s1203 从输入视频流中取下一个视频帧，并且然后在步骤 s1204 矢量预量化数据被编码。接下来，在步骤 s1205 编码/压缩基于非索引的色视频帧。压缩/编码的帧数据在步骤 s1206 被发送到客户机，该数据在步骤 s1207 由客户机接着解码为全色视频帧。在步骤 s1208 现在矢量预量化数据被用于矢量后量化，并且最后客户机在步骤 s1209 修饰视频帧。该处理返回步骤 s1202 处理接着的流中的视频帧。矢量预量化数据包括尺寸为  $32 \times 64 \times 32$  的 3 维阵列，其中，阵列中各单元含有 r、g、b 座标的索引值。十分清楚，存储和发送  $32 \times 64 \times 32 = 65536$  索引值是在技术上不实际的大的额外开销。解决办法是按紧凑的表示编码这个信息。一种方法是如图 30 的从步骤 s1301 开始的流程图所示，利用 8 元树表示编码这个 3 维索引阵列。图 47 的编码器 50 可以使用这个方法。在步骤 s1302，从输入源读出 3D 数据集/视频帧，使得  $F_j$  (r、g、b) 代表 RGB 色空间中对于视频帧中所有 j 像素的所有唯一色。接着在步骤 s1303 中选择最代表 3D 数据集  $F_j$  (r、g、b) 的 N 个编码簿矢量  $V_i$ 。在步骤 s1304 产生 3D 阵列  $t[0 \dots R_{\max}, 0 \dots G_{\max}, 0 \dots$

$B_{\max}]$ 。对于阵列  $t$  中的所有单元，在步骤 s1305 确定最接近码本矢量  $V_i$ 。在步骤 s1306，用于每个单元的最接近编码簿矢量被存储在阵列中。在步骤 s1307，如果以前的视频帧已经被编码，使得以前的阵列  $t$  存在，则步骤 s1308 确定当前与以前阵列  $t$  之间的差，接着在步骤 s1309 产生更新的阵列。然后，在步骤 s1309 更新阵列或者在步骤 s1310 利用有损耗 8 元树方法编码全阵列。这个方法取 3D 阵列（立方体）并且按类似于以 4 元树为基础的表达递归地分裂该阵列。因为矢量编码簿（ $V_i$ ）/色图是自由地动态变化，这种映射信息还进行更新，逐帧地反映这个变化到色图中。建议一种有条件补充方法，利用索引值 255 代表应该未变化座标映射和其它各个值代表 3D 映射阵列的更新值，执行这种操作。类似空间编码，该处理利用预排序 8 元树遍历方法编码色空间映射到色表中。各个透明叶指示叶指示的色空间的区未变化并且索引叶含有由该单元的座标规定的色的色表索引。8 元树编码器在树的顶部开始并且如果该节点是叶的话对每个节点存储单一的 1 比特，或如果是父体的话存储 0 比特。如果是叶并且色空间区是不变的，则存储另外单一的 0 比特，否则对应的色图索引按照  $n$  比特码字进行编码。如果该节点是父节点并且曾存储 0 比特，则 8 个子节点的每个被按所描述的递归地进行存储。当编码器达到树的最低级别，则所有节点是叶节点并且不使用叶/父体指示比特，替代存储接着色索引码字的第一未变化比特。最后，在步骤 s1311，编码的 8 元树被发送到解码器用于后量化数据并且在步骤 s1312 代码簿矢量  $V_i$ /色图被发送到解码器，因此在步骤 s1313 结束矢量预量化处理。解码器执行反向处理、矢量后量化，如图 30 在步骤 s1401 开始的流程图所示。在步骤 S1402 读出压缩的 8 元树数据，并且解码器再生在 s1403 来自编码的 8 元树的 3D 阵列，与描述过的 2D 4 元树解码处理一样。然后，对于任何 24 位色值，对应的色索引可以通过简单查找存储在 3D 阵列最的索引值予以确定，正如在步骤 s1404 表示的。在步骤 s1405 矢量后量化处理结束。这种技术可以用于映射任何非静止 3D 数据为单维数据。当矢量量化被用于选择将被用于代表原始多维数据集的代码簿时，一般这

是应该要求。在哪个阶段执行处理矢量量化并不是问题。例如，我们可能直接 4 元树接着 VQ 编码 24 位数据，或者我们可能首先 VQ 数据并然后 4 元树编码结果，像我们在这里作的那样。这种方法的最大优点是在不同种的环境中，可以允许 24 位数据将被发送到如果能够显示 24 位数据的各个客户机，但是如果不能显示，则可以接收预量化数据并应用这个数据实现实时的 24 位源数据的高质量量化。

图 18 的场景/目标控制数据部分 14 允许每个目标将与一个视频数据流、一个音频数据流和任何其它数据流相关。它还允许对每个目标的各种修饰和表达参数将被贯穿整个场景从此时到彼时动态地修改。这些包括目标的透明、目标的规格、目标的音量、目标在 3D 空间的位置、和目标在 3D 空间的取向（旋转）。

压缩的视频和音频数据现在被传送或存储，用于以后作为一系列数据包传输。存在着多个不同包类型。每个包包括一个公共基本首端和一个有效负荷。基本首端识别包类型、包含有效负荷的包的总尺寸、它与哪个目标相关、和序列识别符。当前定义如下包的类型：**SCENEDEFN**、**VIDEODEFN**、**AUDIODEFN**、**TEXTDEFN**、**GRAFDEFN**、**VIDEODAT**、**VIDEOKEY**、**AUDIODAT**、**TEXTDAT**、**GRAFDAT**、**OBJCTRL**、**LINKCTRL**、**USERCTRL**、**METADATA**、**DIRECTORY**、**VIDEOENH**、**AUDIOENH**、**VIDEOEXTN**、**VIDEOTRP**、**STREAMEND**、**MUSICDEFN**、**FONTLIB**、**OBJLIBCTRL**。如上所述，有 3 种主要类型的包：定义、控制和数据包。控制包（CTRL）被用于定义目标的修饰变换、动画和将由目标控制引擎执行的动作、交互目标的行为、动态媒体组合参数和任何上述执行或应用的条件，这些定义是对于单个目标或者对于正在收视的整个场景的。数据包含有构成每个媒体目标的压缩信息。格式定义包（DEFN）传送配置参数到每个编解码器，并且规定各个媒体目标的格式和如何翻译相关的数据包两者。场景定义包定义场景的格式、规定目标的数量、和定义其它场景的特性。**USERCTRL** 包用于利用基干信道传送用户的交互和返回远端服务器的数据，**METADATA** 包含

有关于视频的元数据，**DIRECTORY** 包含有帮助随机接入比特流的信息，和 **STREAMEND** 包划分流的边界。

### 接入控制和识别

面向目标视频系统的另外的部分是加密/解密用于内容安全性的视频流的装置。通过利用 **RSA** 公共密钥系统编码，将执行解密的密钥被单独和安全地传送到终端用户。

另外的安全措施包括通用的在编码视频流上加唯一商标/识别符。这种措施采取 4 种主要形式：

- a. 在电视会议应用中，一个单一唯一识别符被施加到编码视频流的所有实例，
- b. 在每个视频数据流中有多视频目标的广播视频点播(VOD)中，每个单独视频目标具有用于特定视频流的唯一识别符，
- c. 无线的、超薄客户机系统具有唯一识别符，识别用于该无线超薄系统服务器编码的编码器类型，以及识别这种软件编码器的唯一实例。
- d. 无线超薄客户机系统具有唯一识别符该客户机解码器实例，以便匹配以互联网为基础的用户状况，确定该相连的客户机用户。

唯一地识别视频目标与数据流的能力是特别有益的。在电视会议应用中，没有实际需要监视或记录电话会议的视频数据流，除了在有广告内容出现的情况（唯一地按照 VOD 识别）外。客户机侧解码器软件记录收视的解码视频流（识别符、持续期）。按照实时地或按照连续同步地，这个数据被传送到以互联网为基础的服务器。这个信息被用于产生市场调查税收流以及结合客户机个人概况的市场调查/统计。

在 VOD 中，解码器可以被限制解码广播流或仅有安全密钥时才能使用。当访问互联网认证/接入/帐单服务提供商时，它提供使能解码器授权付费的装置，如果连接到互联网实时地，或按装置以前的同步，可以执行操作使能。另外一种方案，可以对以收视的视频流进行

付费。在类似于广告视频流的电视会议中，随着正在收视期间，解码器记录有关 VOD 的编码视频流。这个信息被传送回互联网服务器，用于市场调查/反馈和付费的目的。

在无线超薄客户机（NetPC）应用中，通过加唯一识别符到编码的视频流中，实现来自互联网或基于计算机服务器的别的地方的视频流的实时编码、传输和解码。客户机侧解码器被使能，以便解码视频流。客户机侧解码器的使能发生在 VOD 应用的沿经授权付费的线路，或者通过安全加密密钥处理，使能接入编码视频流的无线 NetPC 的各种级别。计算机服务器编码软件简化多接入级别。在广播形式中，无线互联网包括从客户机解码器软件反馈到计算机服务器，通过解码器确认监视客户机连接的机制。这些计算机服务器监视服务器应用处理的客户机利用率和相应的变化，并且还监视流向终端客户机的广告。

### 交互音频视频标记语言（IAVML）

这个系统的强有力的部分是通过源程序控制音频-视频场景组成能力。利用源程序，通过源程序语言的限制强加于对组成功能的限制。在这种情况下使用的源程序语言是从 XML 标准得到的 IAVML。IAVML 是用于规定目标控制信息的文本形式，该目标控制信号是被编码为压缩的比特流。

在某些方面 IAVML 类似 HTML，但是特别设计将用于面向目标多媒体时-空空间的，诸如音频/视频。它还可以被用于定义这些空间的逻辑和布局结构，包括分层，它还可以被用于定义链接、寻址并且还有元数据。这是通过允许提供描述和参考信息等的标记的 5 个基本类型实现的。这些是系统标记、结构定义标记、表示格式、和链路和内容。类似 HTML，IAVML 是情况不敏感的，并且每个标记进入打开和关闭形式，用于包封被进行注释的文本的各部分。例如：

<TAG>some text in here</TAG>

音频-视频空间的结构定义使用结构标记并包括如下：

<SCENE>	定义视频场景
<STREAMEND>	在场景中划分流
<OBJECT>	定义目标举例
<VIDEODAT>	定义视频目标数据
<AUDIODAT>	定义音频目标数据
<TEXTDAT>	定义文本目标数据
<GARFDAT>	定义矢量目标数据
<VIDEODEFN>	定义视频数据格式
<AUDIODEFN>	定义音频数据格式
<METADATA>	定义关于给定目标的元数据
<DIRECTORY>	定义目录目标
<OBJCONTROL>	定义目标控制数据
<FRAME>	定义视频帧

由这些结合目录的标记和元数据标记定义的结构允许灵活地访问和浏览面向目标的比特流。

音频-视频目标的布局定义使用基于目标控制的布局标记（修饰参数），以定义在任何给定场景中的各个目标的时空布局，并且包括如下的标记：

<SCALE>	视频目标的比例
<VOLUME>	音频数据的音量
<ROTATION>	3D 空间目标的取向
<POSITION>	3D 空间目标的位置
<TRANSPARENT >	视频目标的透明度
<DEPTH>	改变目标的 Z 序
<TIME>	场景中目标的开始时间
<PATH>	从开始到结束时间动画路径

音频-视频目标的表示定义使用表示标记定义目标的表示（格式定义）并包括如下标记：

<SCENSIZE>	场景空间尺寸
<BACKCOLR>	场景背景色
<FORECOLR>	场景前景色
<VIDRATE>	视频帧速率
<VIDSIZE>	视频帧尺寸
<AUDRATE>	音频取样率
<AUDBPS>	按比特的音频取样尺寸
<TXTFONT>	使用的文本字型
<TXTSIZE>	使用的文本字体尺寸
<TXTSTYLE>	文本风格（黑体、下画线、斜体）

目标行为和动作标记封装目标控制并包括如下类型：



<JUMPTO>	取代当前场景或目标
<HYPERLINK>	设置超链路目标
<OTHER>	重新定标控制另外目标
<PROTECT>	限制用户交互
<LOOPCTRL>	循环目标控制
<ENDLOOP>	暂停循环控制
<BUTTON>	定义按钮动作
<CLEARWAITING>	终止等待动作
<PAUSEPLAY>	播放或者暂停视频
<SNDMUTE>	静音通/断
<SETFLAG>	设置或复位系统标志
<SETTIMER>	设置定时器值和开始计数
<SENDFORM>	发回系统标志到服务器
<CHANNEL>	改变收视信道

文件中的超链路参考允许点击目标调用定义的各动作。

利用具有 **BUTTON**、**OTHER** 和用 **OTHER** 参数定义的 **JUMPTO** 标记定义的多媒体目标可以产生简单的视频菜单，指示当前场景和指示新场景 **JUMPTO** 参数。通过定义 **OTHER** 参数指示背景视频目标和 **JUMPTO** 参数指示替代视频目标可以产生持久的菜单。通过放弃和使能单个的选择，下面定义条件的变化可以用于定制这些菜单。

利用具有从 2 帧视频目标产生的多个检查框的场景可以产生寄存用户选择的简单形式。对每个检查框目标，定义 **JUMPTO** 和 **SETFLAG** 标记。**JUMPTO** 标记用于当该目标被选择或不被选择的情况下，对于指示的该目标选择显示哪一帧图象，并指示的系统标记寄存该选择的状态。利用 **BUTTON** 和 **SENDFORM** 定义的媒体目标可以用于返回该选择到服务器，用于存储或处理。

在可能有多个信道正在广播或多点传送的情况下，**CHANNEL** 标

记能使在单点传送模型操作和广播或多点传送模型和返回之间转换。

各个条件在它们在客户机上执行之前，可以被应用到行为和动作上（目标控制）。这些是利用<IF>或<SWITCH>标记产生条件表达式，按 IAVML 实现的。客户机的条件包括如下类型：

<PLAYING>	当前视频播放？
<PAUSED>	当前视频暂停？
<STREAM>	来自远端服务器的码流
<STORED>	从本地存储器播放
<BUFFERED>	缓冲目标帧#？
<OVERLAP>	将需要拖到哪个目标
<EVENT>	要发生哪个用户事件？
<WAIT>	等待将为真的条件？
<USERFLAG>	是否设置规定用户标志？
<TIMEUP>	定时器超时？
<AND>	用于产生表达式
<OR>	用于产生表达式

控制动态媒体成分处理的可以应用到远端服务器控制的条件包括如下类型：

<FORMDATA>	用户返回的形式数据
<USERCTRL>	发生用户交互事件
<TIMEODAY>	是否是规定的时间？
<DAYOFWEEK>	是周几？
<DAYOFYEAR>	是特定的一天？
<LOCATION>	客户机的地理位置在哪里？
<USERTYPE>	用户人口统计学分类
<USERAGE>	用户的年龄（范围）？
<USERSEX>	用户的性别（男/女）？
<LANGUAGE>	优先的语言？
<PROFILE>	用户概况数据的其它情况
<WAITEND>	等待当前流结束
<AND>	用于产生表达式
<OR>	用于产生表达式

一般 IAVML 文件将有一个或多个场景和一个源程序。按如下方式每个场景定义为有确定的空间尺寸、缺省背景色和任选的背景目标：

```

<SCENE= " sceneone" >
    <SCENESIZE SX= " 320" 、SY= " 240" >
    <BACKCOLR= " #RRGGBB" >
    <VIDEODAT SRC= " URL" >
    <AUDIODAT SRC= " URL" >
    <TEXTDAT> 这是一些文本串</a>
</SCENE>

```

另外一种方式，背景目标可以被事先定义并然后刚好在场景中表示：

```

<OBJECT= " backgrnd" >
    <VIDEODAT SRC= " URL" >
    <AUDIODAT SRC= " URL" >
    <TEXTDAT> 这是一些文本串</a>
    <SCALE= " 2" >
    <ROTATION= " 90" >
    <POSITION=XPOS= " 50" YPOS= " 100" >
</OBJECT>
<SCENE>
    <SCENESIZE SX= " 320" , SY= " 240" >
    <BACKCOLR= " #RRGGBB" >
    <OBJECT= " backgrnd" >
</SCENE>

```

场景中可以含有任何数量的前景目标:

```

<SCENE>
    <SCENESIZE SX= " 320" , SY= " 240" >
    <FORECOLR= " #RRGGBB" >
    <OBJECT= " foregnd_object1" 、 PATH= " somepath" >
    <OBJECT= " foregnd_object2" 、 PATH= " someotherpath"
>
    <OBJECT= " foregnd_object3" 、 PATH= " anypath" >
</SCENE>

```

定义场景中每个动画目标的路径:

```

<PATH= " somepath" >
    <TIME START= " 0" 、 END= " 100" >

```

```
<POSITION TIME=START、XPOS= “ 0”、YPOS= “ 100”  
>  
<POSITION TIME=END、XPOS= “ 0”、YPOS= “ 100” >  
<INTERPOLATION=LINEAR>  
</PATH>
```

利用 IAVML，内容产生器可以文本地产生面向目标视频的动画源程序并且连续地定义动态媒体组合和修饰参数。产生 IAVML 文件以后，远端服务器软件处理 IAVML 源程序，产生可以插入传送到媒体播放器的组成视频流中的目标控制包。服务器还利用 IAVML 源程序内在地知道如何响应通过用户交互从客户机经用户控制包返回传递的动态媒体组合的请求。

### 码流纠错协议

在无线码流的情况下，使用适合的网络协议保证视频数据被可靠地通过无线链路发送到远端监视器。这些可以被诸如 TCP 之类的连接，或者诸如 UDP 之类的无连接取向。协议的特性将取决于所使用的无线网的特性、带宽、和信道特性。该协议执行如下功能：差错控制、流量控制、分组化、连接建立、和链路管理。

有许多现存的为数据网设计的用于这些目的的协议。但是，在视频的情况下，特别关注可能要求对差错的处理，因为，由于视频特性对发送数据的接收和处理的实时限制，有差错的数据的再发送是不合适的。

为了处理这种情况，提供如下的纠错方案：

(1) 视频数据的帧被单个地发送到接收机，每个帧具有添加的能使接收机评估帧中是否含有差错的检验和或循环冗余检验；

(2a) 如果没有差错，则该帧被正常地处理；

(2b) 如果该帧有差错，则该帧被丢弃并且将一个状态消息发送到发送机，指示有差错的视频帧的号；

(3) 当接收到这种差错状态消息时, 视频发送机停止发送所有预定帧, 并代之以立即发送下一个可用关键帧到接收机;

(4) 发送关键帧以后, 发送机重新开始发送正常帧间编码的视频帧, 直至接收到另外的差错状态消息。

关键帧是仅进行帧内编码而不进行帧间编码的一种视频帧。帧间编码是在执行预测处理处并使这些帧取决于所有在先视频帧, 在最后的帧间编码之后并包括其。关键帧是作为第一帧和每当出现差错时发送的。第一帧需要是一个关键帧, 因为没有以前帧用于帧间编码。

### 语音命令处理

因为无线装置是很小的, 对于操作该装置手动输入文本命令和处理数据的能力是困难的。已建议语音命令作为该装置不用手操作的一种途径。但这样出现一个问题, 即在许多无线装置具有非常低的处理能力, 极低于一般的自动语音识别 (ASR) 的要求。在这种情况下解决办法是在该装置捕捉用户的语音, 将其压缩, 并将其发送到服务器用于 ASR 并且按如图 31 所示执行, 因为在任何情况下, 服务器将操作所有用户命令。这使该装置不必须执行这种复杂的处理, 因为它有可能投入其大多数处理资源去解码和修饰任何码流音频/视频内容。这种处理由在步骤 s1501 开始的图 31 的流程图予以描述。在步骤 s1502, 当用户讲一个命令被输入到该装置的麦克风时, 该处理被初始化。在步骤 s1503, 如果语音命令不可用, 则该语音命令被忽略并且处理在步骤 s1517 结束。否则, 在步骤 s1504 语音命令被捕捉和压缩, 在步骤 s1505 编码的样值被插入到 USERCTRL 包, 并且在步骤 s1506 发送到语音命令服务器。然后在步骤 S1507, 语音命令服务器执行自动语音识别, 并且在步骤 s1508 映射转录的语音到一个命令集。在步骤 s1509, 如果该转录的命令不是预定的, 则在步骤 s1510 转录的测试串被发送到客户机, 并且客户机插入文本串到适当的文本字段。如果(步骤 s1509) 转录命令是预定的, 则在步骤 s1512 检查命令类型(服务器或客户机)。如果命令是服务器命令, 则在步骤 s1513 传送到服务器,

并且在步骤 s1514 服务器执行该命令。如果命令是客户机命令，则在步骤 s1515 该命令被返回客户机，并且在步骤 s1516 客户机执行该命令，在步骤 s1517 结束语音命令处理。

## 应用

### 超薄客户机处理和计算服务器

利用超薄客户机作为从任何其它类型个人移动计算装置控制任何类型的计算机的装置，产生虚拟计算网络。在这种新应用中，用户的计算装置不执行数据处理，而作为用户的服务器接口到虚拟计算网络。所有数据处理由位于网络中的计算服务器执行。几乎是，该终端被限制为解码所有输出和编码所有输入数据，包括实际用户接口显示。从结构上讲，输入和输出数据流在用户终端中总的是独立的。在输入数据进行处理的处理服务器中执行输出或显示数据的控制。因此，图形用户接口（GUI）解压输入为 2 个独立的数据流：视频输入和输出显示部分。输入流是命令序列，可用是 ASCII 字符和鼠标或笔事件的组合。对于大的范围，解码和修饰显示数据包括这种终端的主要功能，并且可用修饰复杂的 GUI 显示。

图 32 表示操作在无线 LAN 环境的超薄客户机系统。这种系统可能同样地工作在无线 WAN 环境，诸如 CDMA、GSM、PHS 或类似的网络。在无线 LAN 环境系统中，从 300 米的户内到高达 1km 户外的范围是典型的范围。超薄客户机是具有无线网卡和接收信号的天线的个人数字助理或掌上计算机。无线网卡通过 PCMCIA 槽、紧凑闪存口或其它装置接口个人数字助理。计算服务器可用是运行 GUI 的任何计算机，GUI 连接到具有无线 LAN 能力的互联网或局域网。计算服务器系统可以由执行 GUI 程序（11001）组成，该程序通过具有包含音频和 GUI 显示的程序输出的客户机响应（11007）进行控制，利用程序输出视频变换器（11002）进行读出和编码。通过 11002 中的第一视频编码 GUI 显示到远端控制系统（11012）的传送，该系统利用 OO 视频编码（11004）变换通过 GUI 屏读出（11003）的 GUI 显示，和

通过音频读出 (11004) 的任何音频, 利用以前描述的处理压缩视频以便编码和发送它到超薄客户机。GUI 显示可以利用 GUI 屏读出 (11003) 进行捕捉, 它是在诸如 Microsoft Windows 中的 CopyScreenToDIB ( ) 之类的许多操作系统中的标准功能。超薄客户机经 Tx/Rx 缓冲器 (11008 和 11010) 接收压缩的视频并在经 O O 视频解码 (11010) 解码以后, 利用 GUI 显示和输入 (11009) 对用户显示适当地修饰。任何用户控制数据被发送回计算服务器, 在服务器中利用超薄客户机对 GUI 控制翻译 (11006) 进行翻译, 并且通过程序性 GUI 控制执行 (11005) 控制执行 GUI 程序 (11001)。这包括执行新的程序、结束程序、执行操作系统功能、和与运行程序有关的任何其他的功能。这种控制在各种形式下都是有效的, 在 M S Window NT 的情况下, 可以使用 Hooks / Journal Playck Fune ( )。

对于较大范围的应用, 最好是图 33 的 WAN 系统。在这种情况下, 计算机服务器直接连接到标准电话接口。发送 (11116) 用于通过 CDMA、PHS、GSM 或类似蜂窝电话网。在这种情况下超薄客户机包括具有连接到电话的个人数字助理、手持机和调制解调器(11115)。所有其他方面类似于描述在图 32 的 WAN 系统的配置。在这种系统的变形中 PDA 和电话集成在一个装置中。在这种超薄客户机系统的一个例子中, 移动装置从任何位置全面接入到计算服务器, 同时达到诸如 CDMA、PHS 或 GMS 之类的标准移动电话网。这种系统不用移动电话的有电缆的方式也可以使用, 使得超薄计算装置通过调制解调器被直接连接到有线网络。

计算机服务器还可以位于远端并经互联网或内联网 (11215) 连接到本地无线发送机/接收机 (11216), 如图 34 所示。这种超薄客户机应用特别适合于应急互联网的进程活动关系为基础的虚拟计算系统。

### 丰富音频-视频接口

在没有目标控制数据插入到比特流中的超薄客户机系统中, 客户机可以不执行处理, 而是修饰单一视频目标进行显示并且返回所有用



户交换到服务器用于处理。虽然该方法可以被用于访问远端执行处理的图形用户接口，但它不适合用于产生本地执行处理的用户接口。

规定 DMC 的目标为基础的能力和交换引擎，这个整个系统和它的客户机-服务器模型特别适合用于丰富音频-视频接口的核芯。不像基于大致静止图符和矩形窗口的典型图形用户接口一样，当前的系统利用多视频和其它媒体目标能够产生丰富用户接口，这些目标可以简化与本地装置或远端程序执行的设施进行交互。

### 多方无线电视会议处理

图 35 表示包含 2 个或多个无线客户机电话装置的多方无线电视会议系统。在这种应用中，2 个和多个参与者在他们当中可以被设置多个视频通信链路。存在非集中控制机制，并且每个参与者可以决定在多方会议中哪些链路激活。例如，在包括 A、B、C 三人的三人会议中，可以在人 AB、BC 和 AC 之间形成链路（3 个链路），或者可替代地是 AB 和 BC 但没有 AC（2 个链路）。在这种系统中，按照他们的意愿，每个用户可以同时设置许多链路到不同的参与者，不要求集中网络控制并且每个链路是分别管理的。对于每个新的电视会议链路的输入视频数据形成一个新的视频流，该流被馈送到连接到相关于输入视频数据的链路上的每个无线装置的面向目标解码器。在这种应用中，目标视频解码器（面向目标解码器 11011）按照一种表现模型被运行，在该模型中每个视频目标基于正在显示的多个视频目标，按照布局规则进行修饰。各个视频目标之一可以按照当前动作被识别，并且这个目标可以在比其他目标大的尺寸上被修饰。利用具有最大声音能量（响度/时间）的视频目标的自动装置或者由用户手动可以执行哪个目标是当前动作的选择。客户机电话装置（11313、11311、11310、11302）包括个人电话助理、手持个人计算机、个人计算装置（诸如笔记本和台式 PC）和无线电话手持机。客户机电话装置可以包括无线网卡（11306）和发送和接收信号的天线（11308）。无线网卡通过 PCMCIA 槽、紧凑闪存口或其它连接接口接口客户机电话装置。无线电话手机

可以用于 PDA 无线连接 (11312)。在 LAN/互联网/内联网之间可以建立一个链路 (11309)。每个客户机装置 (例如, 11302) 可以包括数字视频捕捉的摄象机 (11307) 和用于声音捕捉的一个或者多个麦克风。客户机电话装置包括视频编码器 (OO 视频编码 11305), 利用上述的处理压缩捕捉的视频和音频信号, 这些信号然后被发送到一个或者多个其他客户机电话装置。数字摄象机可以仅捕捉数字视频和传送它到客户机电话装置用于压缩和传送, 或还可以利用 VLSI 硬件芯片 (ASIC) 压缩视频本身并传送编码的视频到用于发送的电话装置。含有特殊软件的客户机电话装置接收压缩的视频和音频信号并利用上述处理适当修饰它们到用户显示和麦克风输出。这个实施例还包括在客户机电话装置上利用上述交互目标管理的处理, 直接视频管理或做广告, 这可以通过与上述其他客户机电话装置相同的装置被反映 (在 GUI 显示器上重现), 特别是在相同的电话会议中。这个实施例可以包括在客户机电话装置之间用户控制数据的发送, 例如对其它客户机电话装置提供遥控。任何用户控制数据被发送回适当的客户机电话装置, 进行翻译和然后用于控制视频图象和其他软件和硬件功能。作为在超薄客户机系统的应用的情况, 可以使用各种网络接口。

#### 有定标图象用户做广告的按需交互动画或视频

图 36 是有目标用户视频广告按需系统的交互视频的方框图。在这个系统中, 服务提供商 (例如, 生活新闻、视频点播 (VOD) 提供商等) 将单点传送或者多点传送视频数据流到单个用户。视频广告可以包括来自不同源的多个视频目标。在视频解码器的一个例子中, 小的视频广告目标 (11414) 被动态地组成为视频流, 传送到解码器 (11404) 将被修饰为在某种时间进行收视的场景。这个视频广告目标可以来自存储在库 (11406) 中装置中的预下载的广告, 或者经在线视频服务器 (例如, 视频按需服务器 11407) 来自远端存储器发来的码流 (11412) 进行改变, 能够利用视频目标叠加 (11408) 进行动态媒体组合。这个视频广告目标根据该客户机自己 (用户的) 概况信息有目标地特别针

对该客户机装置(11402)。用户的概况信息可以具有存储在多个位置的各个部分,例如在线服务器库(11413)或本地客户机装置。对于基于广告的有目标视频,利用视频流的反馈和控制机制以及其收视。服务提供商或者另外一方可以保持和操作存储压缩视频流(11412)的视频服务器。当用户从视频服务器选择一个节目时,提供商的发送系统自动从用户概况数据库(11413)获得的信息中,选择哪个促销或广告数据是可应用的,该数据库可以包括诸如用户年龄、性别、地理位置、预定历史、个人喜好、购买历史等。然后,可以作为单一视频目标存储的广告数据连同请求的视频数据一起可以插入发送的数据流中并且发送给用户。作为单独的视频目标,然后,通过调整表现/显示特性,用户可以与广告视频目标进行交互。通过对目标点击或拖等操作,用户还可以与广告视频目标进行交互,因此发送消息返回视频服务器,指示用户希望激活某些与广告视频目标相关的功能,作为服务提供商或广告目标提供商确定的目标。这种功能可以简单地要求从广告商对另外信息的请求,设置视频/电话呼叫给广告商,初始化销售凭证处理,根据事务处理或者某些其他控制形式初始化近程。除了广告外,这个功能可以由服务提供商直接使用,促进附加的视频贡献,诸如其他可用视频信道、它可以按小的移动图符图象做广告。在这种情况下,在这种图符上点击的用户动作可以通过提供商使用,以改变发送到用户的主要视频数据或者发送附加数据。可以通过视频目标的重叠(11408)组合多个视频目标数据流到最后的组合数据流,它被发送到每个客户机。通过按照以前描述的实时或预处理编码(视频编码 11411)的来自诸如其他视频服务器、网络摄象机(11410)、或者计算服务器之类的不同远端源的视频促进选择(11409),被组合的分别的视频目标流的每个可以通过互联网进行检索。再有,作为超薄客户机和电视会议的其他系统应用,可以使用各种优选的网络接口。

有图象广告的一个实施例中,视频广告目标可以类似如图 37 所示的按钮那样进行编程操作,当由用户选择时,可以做如下事情之一:

- 立即改变正在收视的视频场景跳到新的场景,该场景提供更多关

于正在做广告的产品信息，或者到能够存储的在线电子商务。例如，可以用于改变“视频信道”。

- 通过利用另外的目标替代该目标，提供更多关于正在做广告的产品信息，立即改变视频广告目标为类似子题目的码流文本信息。这在显示的场景中不影响任何其他视频目标。

- 去掉视频广告目标并设置指示用户已经选择了广告的系统标志，然后当前视频将正常地播放到结束并且然后跳到指示的广告目标。

- 发送消息返回服务器，寄存感兴趣的产品以进一步异步跟踪信息，可以通过经电子函件或作为附加码流视频目标等。

- 在视频广告目标仅被用于做标记的目的场合，点击目标可以触发它的不透明并使它半透明，或者能使它执行预定的动画，例如按 3D 旋转或在圆形路径上移动。

利用视频广告目标的另外的方式是对移动智能电话用户补贴分组费用或者呼叫费用，通过如下方式；

- 对于无条件响应呼叫期间或呼叫结束期间，自动显示响应者的视频广告目标。

- 如果用户与目标执行某些交互，在提供发起呼叫之前、期间、或之后，显示交互视频目标。

图 37 表示有图象广告系统的一个例子，当有图象广告会话开始（内流广告开始 S1601）时，对于音频-视频流（从服务器请求 AV 数据流 S1602）的请求从客户机装置（客户机）被发送到服务器处理。服务器处理（服务器）可以位于客户机装置或者远端在线服务器。响应于该请求开始发送码流请求数据（S1603）到客户机。在码流数据由客户机接收的同时，它执行修饰该数据流的处理，并且接受和响应用户的交互。因此，客户机检查是否接收的数据指示当前 AV 码流的结束已经达到（S1604）。如果这是真，并且除非存在另外排队的 AV 数据流（S1605）是当前码流刚结束的未完码流，则该有图象内广告会话可以结束（S1606）。如果排队 AV 数据流存在，则服务器开始发送新的 AV 数据码流（返回到 S1603）。虽然在数据流的处理中 AV 流

的结束还未达到 (S1604-否) 并且如果当前广告目标未开始发码流, 则服务器可以选择 (S1608) 并根据包括位置、用户概况等各个参数在 AV 流 (S1609) 中插入新的广告目标。如果服务器正在处理发的 AV 数据流并且广告目标已经被选择并且插入到 AV 流中, 则如上所述客户机解码该比特流并且修饰各个目标 (S1610)。在 AV 流可能继续的同时, 由于各种原因, 包括客户机交互、服务器交互、或者广告流结束, 图象内的广告流可能结束 (S1611)。如果图象中广告流已经结束 (S1611-是), 则通过 S1608 可重新选择新的图象内广告。如果 AV 数据流和图象内广告流继续 (S1611-否), 则客户机捕捉与广告目标的任何交互。如果用户点击该目标 (S1612-是), 则客户机发通知到服务器 (S1613)。服务器的动态媒体组合程序的源程序定义哪些动作将要采取作为响应。这些包括: 无动作、延迟 (推迟) 或者立即动作 (S1614)。在无动作 (S1614-无) 的情况下, 服务器可以寄存这个事实, 用于接着以后 (在线或离线) 的动作 (S1619), 这可能包括更新用户的概况信息, 它可能用于定标类似的广告或者接着的广告。在延迟动作 (S1614-延期) 的情况下, 则要采取的动作可能包括寄存 (S1619) 以下采取 S1619 或排队新的 AV 数据 (S1618), 用于发当前未完成 AV 数据流的码流。在当服务器在客户机装置的情况下, 这可以进行排队和当该装置可以下一个被连接到在线服务器时进行下载。在远端在线服务器的情况下, 则当当前 AV 流完成时, 则排队的流可以进行播放 (S1605-是)。在立即动作 (S1614-立即) 的情况下, 则根据加到广告目标上的控制信息可能执行多个动作, 包括对当前广告目标改变动画参数 (S1615-动画)、替换当前广告目标 (S1615-广告) 和替换当前 AV 流 (S1617)。动画请求改变 (S1615-动画) 可能导致对目标的修饰, 例如变换或旋转、和透明度等。这可能在以下的步骤进行寄存 (S1619)。在广告目标改变请求 (S1615-动画) 情况下, 与以前 (S1608) 一样可能选择一个新的广告目标。在另外一个实施例中, 这种视频系统的动态媒体组合能力可能被用于能够使收视者定制他们的内容。一个例子是用户可以能够从一个故事情节的主要人物中选择一

个人物。在这种具有动画卡通情况下，收视者能够从男或女的人物中选择。这种选择可能从共享的人物组中交互地执行，例如在线的多参与者环境或者可以根据存储的用户概况。选择男的人物将产生组合到比特流中的男人物声视媒体目标，替代女的人物。在另外的例子中，不是固定情节选择主要人物，通过在收视期间进行选择，情节本身可以改变，通过选择某个场景跳动显示下一个场景改变情节。在任何给定的点上多个可以替换的场景可以使用。选择可由各种机制限制，如以前的选择、所选择的视频且标以及故事情节中图象的位置。

服务提供商可以提供用户认证和对视频材料的访问控制、内容消费量的计量和使用帐单。图 41 表示所有用户在他们被提供接入服务（例如，内容服务）之前，能寄存认证/接入提供商（11507）系统的实施例。认证/接入服务将对每个用户产生‘唯一识别符’和‘接入信息’（11506）。当客户机在线时（例如，首先接入服务器），唯一识别符将被自动转移动客户机装置（11502）用于本地存储。由用户的所有后续请求经视频内容提供商（11511）存储的视频内容（11510）可以利用客户机系统的用户识别符进行控制。在一个利用率的例子中，用户将被计帐能够由通过他们的唯一识别符认证的用户访问内容有规律签约的费用。另外一种方案，可以通过利用率收集每收视一次付费一次状态的计费信息（11508），关于诸如计量之类的利用率的信息可以由内容提供商进行记录（11511）和馈送到一个或多个计费服务提供商（11509）和接入代理人/计量提供商（11507）。对不同用户和不同内容可以授予不同的接入级别。由以前系统实现的无线接入可以有多种方法实现。图 41 表示通过 Tx/Rx 缓冲器（11505）接入客户机装置（11502）到本地无线发送机（11513），该发送机经 LAN/内联网或不包括无线 WAN 的互联网连接（11513）提供接入到服务提供商。客户机装置可以实时地依靠接入代理人/计量提供商（11507），以增加对内容的接入权限。编码的比特流可以按照以前描述的通过 11504 进行解码并利用以前描述的（11503）客户机交互对场景进行修饰。接入控制和或计费服务提供商可以保持用户的利用率概况，然后它可以出

售或者许可给第三方，用于广告/促销的目的。为了实现计费和利用率控制，如上所述，可以开发适合的加密方法。除此以外，如上所述，可以使用唯一的做标记/识别编码视频的处理。

## 视频广告节目集

可以下载交互视频文件，而不是发码流到一个装置，使得如图 38 所示在任何时间可以收视离线或在线。下载视频文件仍然保持如上所述的由在线码流处理提供的所有交互和动态媒体组合能力。视频节目集可以包括菜单、广告目标、和寄存用户选择和反馈的一致形式。差别仅在于，因为视频节目集可以进行在线收视，加到视频目标上的超链路可以不指定没有位于该装置上的新的目标。在这种情况下，客户机装置可能存储不能从装置中的数据进行服务所有用户的选择并且当下次该装置在线或者与 PC 同步时，传送这些到适当的远端服务器。按这种方式传送用户的选择可以产生将要执行的各种动作，例如提供进一步的信息、下载要求的场景或者链接到要求的 URL。交互视频节目集可用于许多内容类型，例如交互广告节目集、团体训练内容交互娱乐和用于交互货物和服务的在线和离线购买。

图 38 表示交互视频节目集 (IVB) 的一种可能的实施例。在这个例子中，当请求时（从服务器拉出）或者按照事先安排（从服务器推出）（S1701），IVB（SKY 文件）文件数据可以被下载到客户机装置（S1702）。下载可能出现在任何情况下，经由与台式 PC 同步无线地，或按诸如紧凑闪存、或存储棒之类的媒体存储技术分配。客户机播放机将解码比特流（按照以前描述的）并且修饰来自 IVB 的第一个场景（S1703）。如果播放机达到 IVB 的结束（S1705-是），则 IVB 将结束（S1708）。当播放机未达到 IVB 的结束，则它修饰各个场景并且无条件的执行所有目标控制动作（S1706）。用户可以按照目标控制所定义地与目标交互。如果用户不与目标交互（S1707-否），则播放机继续从数据文件中读出（S1704）。如果用户与目标在场景中交互（S1707-是）和目标控制动作是执行形成操作（S1709-是），并且如

果用户是在线（S1712-是），则形成的数据将被发送到在线服务器（S1711），否则如果是离线（S1712-否），则形成数据将被存储，用于当该装置回到在线时以后的加载（S1715）。如果目标控制动作是跳跃行为（S1713-是）和规定跳到新场景的控制，则播放机将寻找新场景在数据文件中的位置（S1710）和继续从那里读数据。如果控制规定跳到另外的目标（S1714-目标），则这将通过访问按照存储在数据文件中的该场景的正确数据流，产生将被替代和被修饰的对象目标（S1717）。如果目标控制动作改变目标的动画参数（S1716-是），则目标动画参数将可能被更新或者取决于由目标控制规定的参数进行动作（S1718）。如果目标的控制动作对该目标执行某些其他操作（S1719-是）和由控制规定的所有条件都满足（S1720-是），则执行控制操作（S1721）。如果选择的目标不具有控制操作（S1719-否、S1720-否），则播放机可以继续读出和进行修饰视频场景。在任何这些情况下，动作的请求可以被记录和通知可以被存储，用于如果离线以后加载到服务器，或者如果在线被直接转移到服务器。

图 39 表示用于广告和购买应用的交互视频节目集的一个实施例。该例子包含用于在线购买和内容收视选择的形式。IVB 被选择和播放开始（S1801）。引入的场景可能进行播放（S1802），该场景由多个目标组成，如所示（S1803、视频目标 A、视频目标 B、视频目标 C）。所有视频目标可能具有由它们附加的控制数据定义的各种修饰参数动画，例如 A、B 和 C 可能在主收视目标已经开始进行修饰以后从右侧移动（S1804）。用户可以与任何目标进行交互并初始化目标控制动作，例如用户可以点击 B（S1805），它可能具有“跳到”超链路，控制动作停止播放当前场景并开始播放由控制参数指示的新的场景（S1806、S1807）。这可能含有多个目标，例如它可能含有用于用户可以选择的导航控制的菜单目标（S1808），以返回到主场景（S1809、S1820）。用户可能与另外目标交互，例如 A（S1811），它可能具有跳到另一规定场景的行为（S1812、S1813）。在所示的例子中，用户可能再次选择菜单选择（S1814）返回主场景（S1815、S1816）。另



外的用户交互可能是拖目标 B 到购买篮中 (S1817)，这可能引起另外的目标控制的执行,该目标控制是对重叠目标 B 和购买篮有条件的,以通过设置适当用户状态标志变量 (S1818) 寄存购买请求并且还根据动态媒体组合,引起目标动画或变化 (S1819、S1820)，在这个例子中表示购物篮已满。用户可能与购物篮目标交互 (S1821)，该交互可能具有跳跃到检查事务处理和信息场景的行为 (S1822、S1823)，这些可表示请求的购买。显示在这个场景的目标将根据用户标志变量的值由动态媒体组合确定。用户可以与各个目标交互,通过按照由目标控制参数的定义修改用户标志,改变它们的购买请求状态通/断,这些将引起动态媒体组合处理表示在场景中已选择目标或者不选择目标。用户可以交替地选择与购买交互,或者返回可能已经跳到以适当场景作为对象的新场景控制行为的目标,例如主场景或者涉及进行事务处理的场景 (S1825)。涉及的事务处理可能被存储在客户机装置,如果离线用于以后加载到服务器或者可能被实时地加载到服务器,如果客户机装置在线用于购买/信用卡确认。选择购买目标可能跳到确认场景 (S1827、S1828)，同时事务处理可能被发送到服务器 (S1826)，在事务处理完成以后 (S1824) 播放任何剩余的视频。

### 分配模型和 DMC 操作

存在多种用于传送比特流到客户机的分配机制,包括:与客户机同步地下载到台式 PC、无线在线连接到装置和紧凑媒体存储装置。内容的传送可以由客户机始发或者由网络始发。分配机制传送始发的组合提供了多种传送模型。一种这样的模型客户机启动传送的是按需发码流,其中一个实施例是提供低带宽和低闲置信道的按需发码流(例如,无线 WAN 连接)并且内容是实时发送到收视发送的码流的客户机装置的。第二种内容传送模型是客户机通过在线无线连接启动传送,例如利用文件转移协议,在播放以前内容可以被迅速全部下载,一个实施例提供高带宽、高闲置信道,其中内容被立即传送和接着收视。第三传送模型是网络启动传送,其中一个实施例提供低带宽和高闲置,

这种装置被称为“始终接通”，因为客户机装置可以始终在线。在这种模型中，视频内容可以被在夜间或全体非高峰期间巧妙地下载到装置中，并且缓冲在存储器中用于以后收视。在这种模型中，系统的操作不同于上面的第二种模型（客户机启动按需下载），其中用户应当寄存向内容服务提供商请求传送特殊内容。然后，这个请求将被用于由服务器自动安排网络向客户机装置的启动传送。当内容传送的适合时间发生时，例如网络利用率的非高峰时，服务器将建立与客户机装置的连接、协商传输参数并管理与客户机的数据转移。另一种方案，利用所在网络已分配的任何可用残留带宽（例如，恒定速率连接），服务器将每次少量发送数据。通过可视或可闻的指示向用户发信令，用户将知道所请求的数据已经全部被传送，使得当用户准备好时，然后他们可以收视数据。

播放机能够处理推或拉两种发送模型。一个系统操作的实施例如图 40 所示。无线码流的会话可以由客户机装置（S1903-拉）或者由网络（S1903-推）（S1901）开始。在客户机启动的码流会话中，客户机可以通过各种方式（S1904）启动码流，例如输入 URL 与交互目标的超链接、或者拨无线服务提供商的电话号码。连接请求可以从客户机发送到远端服务器（S1906）。服务器可以建立和开始拉连接（S1908），它可以发送码流到客户机装置（S1910）。在发码流期间，如上所述客户机进行解码和修饰比特流以及取用户的输入。由于更多数据被发送（S1912-是），服务器继续发新的数据到客户机，用于解码和修饰，如上所述这种处理可以包括交互性和 DMC 功能。一般，当码流中没有更多数据（S1912-否）时，用户可以结束从客户机装置的呼叫（S1915-柱），但是用户可以在任何时间结束呼叫。呼叫的结束将停止无线码流会话，否则如果在数据已经完成发送以后用户不结束呼叫，客户机可以进入空闲状态，但是仍然在线。在网络启动无线码流会话（S1903-推）的例子中，服务器将呼叫客户机装置（S1902）。客户机装置将自动应答呼叫（S1905）与客户机建立推连接（S1907）。建立的处理可以包括服务器与客户机之间协商有关客户机装置的容量、或配置或用

户的具体数据。然后，服务器将发数据到客户机（S1909），客户机存储接收的数据用于以后收视（S1911）。当更多的数据需要发送（S1912-是）时，这个处理将继续通过非常长的时间周期（低带宽缓慢下码流）或者通过较短的时间周期（高带宽下载）。当整个数据流或者某些源程序位置达到码流中（S1912-否），则在这种推连接中（S1915-推）的客户机装置可以发信号给用户，用于播放的内容已经准备好（S1914）。在发完所有需要的内容以后，服务器将结束到客户机装置的呼叫或者连接（S1917）以结束无线码流会话（S1918）。在另外一种个实施例中，利用网络启动的消息给无线客户机，利用发给无线客户机装置的网络启动消息，如上所述当该消息被接收时可以由用户进行交互以开始推连接，推与拉连接之间的混合操作可能发生。按这种方式，通过由含有适合超链路的数据网络安排的传送可以促进推连接。

这 3 种分配模型适合于单点传送操作模式。在如上所述的第一种按需分配模型中，远端码流服务器可以无限制地执行动态媒体组合和实时地处理用户交互并且执行目标控制动作等，而在另外两种模型中，当用户可能收视离线内容时，本地客户机可以处理用户交互和执行 DMC。如果客户机在线，将被发送到服务器的任何用户交互数据和形成的数据可以被立即传送，或者如果用户离线在承担转移数据的后续处理的不确定时间进行传送。

图 42 是描述按照本发明的无线码流播放机/客户机执行按需发无线视频码流的主要步骤的实施例的流程图。客户机应用程序在步骤 S2001 开始，在步骤 S2002 等待用户输入 URL 或者远端服务器的电话号码。当用户输入远端服务器 URL 或者电话号码时，在步骤 S2003 软件初始化与无线网络的网络连接（如果尚未连接）。在连接建立以后，在步骤 S2004 客户机软件请求将从该服务器发码流的数据。然后，客户机继续处理按需发视频码流，直至在步骤 S2005 用户请求断开，软件前进到步骤 S2007 启动与无线网络和远端服务器的呼叫断开。最后，在步骤 S2009 软件释放可能已经分配的各个资源并且在步骤 S2011 客户机应用程序结束。直至用户请求，将在步骤 S2005 结束的呼叫前

进到检查接收网络的数据的步骤 S2006。如果没有数据被接收到，软件返回到步骤 S2005。但是如果从网络接收到数据，则输入数据在步骤 S2008 缓冲，直至接收到整个包。当在步骤 S2010 接收到完全的包时，检查数据包的差错、序列信息、和同步信息。如果在 S2012 该数据包包含差错、或者是在序列之外的，在步骤 S2013，状态消息被发送到远端服务器指示这种情况，接着返回步骤 S2005 检查用户呼叫断开请求。但是如果在 S2012 无差错地接收到包，则前进到步骤 S2014 并且在步骤 S2014 数据包被传送到软件解码器并进行解码。在 S2015 解码的帧在存储器中被缓冲，用于在步骤 S2016 进行修饰。最后，应用程序返回到步骤 S2005，检查用户呼叫断开，无线码流播放机应用程序继续。

除单点传送外，其它的操作模式包括多点传送和广播。在多点传送和广播的情况下，系统/用户交互和 DMC 能力被限制和可能按与单点传送模型不同方式操作。在无线环境中，很可能多点传送和广播数据将在分别的信道中发送。与分组网一样，不存在纯逻辑信道，而代之以这些可能是电路交换信道。一个单一的发送是从一个服务器发送到多个客户机的。因此，用户交互数据利用用于每个用户的单独的单个单点传送 ‘反向信道’ 连接，可以被返回到服务器。多点传送与广播之间的区别是，多点传送数据可能仅在某些地理边界，例如无线小区的范围进行广播。在数据传送到客户机装置的广播模型的一个实施例中，数据可以被发送到一个网络中的所有无线小区，通过特定的无线信道广播数据使客户机装置接收。

可以利用一个广播信道的例子是发送含有服务目录的场景的循环。各个场景可能被分类为含有一组对应于其它选择的广播信道的超链路视频目标，使得用户选择的一个目标将改变相关的信道。另一个场景可以含有一组适合视频按需服务的超链路视频目标，其中用户通过选择一个视频目标，可能产生一个新的单点传送信道并且从广播转换到这个信道。同样，在单点传送按需信道中的超链路目标将能够改变由客户机接收的比特流为从特定的广播信道接收的比特流。

因为多点传送或广播信道从服务器到所有客户机发送相同数据，DMC 被限制在它的为每个用户订制场景的能力。在广播模型中 DMC 对信道的控制可以不受该单个用户的限制，在这种情况下它不可能与单个用户交互修改广播的比特流的内容。因为广播依靠实时码流，相同的方法不可以用于本地客户机 DMC 作为离线收视，在这种情况下每个场景可以具有多个目标流并且可以执行跳跃控制。但是，在广播模型下，用户不能完全禁止与场景交互，他们仍然自由地修改修饰参数，诸如激活动画等，利用服务器寄存目标选择，并且通过激活与视频目标相连的任何超链路进行跳跃，他们自由地选择一个新的单点传送或广播信道。

DMC 可以被用于在广播中订制用户经历的一种方法是监视正在收视该信道的不同用户的分配和根据平均用户的概况构成定义将被修饰的场景的输出比特流，例如图象中的广告目标的选择可以基于收视者主要是男的还是女的。在广播状态 DMC 可以被用于订制用户的经历的另外一种方式是发送具有多个媒体目标组合的比特流，而不考虑当前收视者分配。在这种情况下，客户机根据用户的概况从广告目标中进行选择，产生最终的场景。例如，按照多种语言的多种字幕可以被插入到广播的定义的场景的比特流中。然后，客户机根据特定的条件在比特流中的目标控制数据广播选择哪种语言字幕修饰。

## 视频监视系统

图 43 表示视频监视系统的一个实施例，它可能用于实时地监视许多不同的环境，诸如家庭财产和家族、商业财产和员工、交通、气候和特别感兴趣的地方。在这个例子中，摄象机装置 (11604) 可以被用于视频捕捉。捕捉的视频可能在 11602 按如上所述被编码，如上所述该编码利用组合来自其它存储器 (11602) 或者利用控制 (11607) 从服务器远程发送的码流的附加视频目标的能力。监视装置 (11602) 可能是：摄象机部分（例如按 ASIC 实现的）、客户机装置部分（例如具有摄象机和 ASIC 的 PDA）、分别来自摄象机（例如，分别的监视

编码装置)或者来自远端视频捕捉(例如,具有活动视频馈送的服务器编码处理)。编码的比特流在安排的时间可以进行发送或进行下载到客户机装置(11603),在那里该比特流可以被如上所述地解码(11609)和修饰(11608)。除了通过利用无线 LAN 接口在近距离范围发送远端视频到无线处理装置外,监视装置(11602)还能够利用标准无线网络基础设施通过长距离发送远端视频:例如通过利用 PHS、GSM 或其它无线网络的 TDMA、FDMA、CDMA 传输的电话接口。其它接入网结构也可以使用。该监视系统具有智能功能,诸如移动检测告警、自动通知和拨号告警、记录和检索视频段、多个摄象机输入之间的选择和切换、和在远端位置提供给用户多个数字或模拟输出。这种应用包括家庭安全、儿童监护和交通监视。在这个最后的情况下,活动交通视频被发送给各个用户并且可以按多种可替代的方法执行:

a. 用户拨号特定的电话号码和然后选择交通摄象机位置,收视由操作员/交换机进行处理的范围。

b. 用户拨号特定电话号码和用户的地理位置(例如从 GPS 或者 GSM 小区三角定位得到的)被用于自动提供交通摄象机位置的选择,收视可能伴随的交通信息。在这种方法中用户任选地规定他或者她的目的地,可以用于帮助提供交通摄象机的选择。

c. 用户可以寄存特殊的服务,服务提供商将呼叫该用户和自动发送码流视频,显示可能具有潜在塞车的汽车路由。当进行寄存时,用户可以决定一个或多个为此预安排的路由,这些路由可以被系统存储,在与来自 GPS 或者小区三角定位的定位信息的配合下帮助预测用户的路由。系统将跟踪用户的速度和位置确定行驶和正在遵循的路由方向,然后潜在的路由搜索它的监视交通摄象机表,确定是否有地方塞车。如果有,则系统将通知驾驶员有塞车并且向用户提供最相关的交通图。静止的用户或以步行速度行进的用户将不被呼叫。另外规定指示塞车的交通摄象机,系统可以通过寄存用户表搜索并告警他们,所述用户正行驶在那里。

## 电子贺卡服务

图 44 是用于智能移动电话 11702 和 11712 和无线连接的 PDA 的电子贺卡服务的一个实施例的框图。在这个系统中，一个开始的用户 11702 可以利用互联网连接的个人计算机 11707 的互联网 11708 或者利用移动智能电话 11706 或者无线连接 PDA 的移动电话网 11703 访问贺卡服务器 11710。贺卡服务器 11710 通过软件接口，允许用户从存储的服务器中的模板库 11711 选择订制的贺卡模板。模板是一段短视频或者动画，包括多个诸如：生日祝愿、明信片、良好祝愿等内容。订制可能包括文本和或音频内容插入到视频和动画模板中。在订制以后，用户可以付给用于事务处理和传送电子贺卡到个人的移动电话号码的费用。然后，电子贺卡被传送到码流服务器 11712，将被存储器。最后，贺卡经无线电话网 11704 在非高峰期间从码流媒体服务器 11709 传送到希望的用户 11705 的移动装置 11712。在明信片的情况下，在每个地理位置用于移动电话网的特定模板视频可以被产生，仅可以员物理地在该当地的人进行发送。在另外的实施例中，用户能够加载一个短的视频到远端应用服务提供商，然后该提供商压缩该视频和进行存储，用于以后传送到目的地的电话号码。图 45 是表示按照本发明用户可能执行产生并发送电子贺卡的一个实施例的主要步骤的流程图。处理在步骤 S2101 开始，在该步骤经互联网或者无线电话网用户被连接到应用服务器提供商 ASP。在步骤 S2102，如果用户希望使用他们自己的视频内容，用户可以捕捉活动视频或者从多个源的任何一个源获得视频内容。在步骤 S2103，这个视频内容被存储在一个文件中，并且在步骤 S2105 由用户对应用服务提供商进行更新并由贺卡服务器进行存储。在步骤 S2102 如果用户不希望使用他们自己的视频，则前进到步骤 S2104，在该步骤用户从由 ASP 保持的模板库选择贺卡/电子函件，在步骤 S2106 用户可选择订制视频贺卡/电子函件，因此在步骤 S2107 用户从模板库选择一个或者多个视频目标，并且在 S2108 应用服务提供商插入选择的目标到已选择的视频数据中。当用户已经完成订制的电子贺卡/电子函件时，在步骤 S2109 用户输入目的地电话号码

/地址。接下来在步骤 S2110 ASP 压缩数据流并且存储它用于传送到码流媒体服务器。在步骤 S2111 处理完成。

### 无线本地环路码流视频和动画系统

另外的应用是无线接入到存储在本地服务器的包含音频-视频训练材料，或者用于无线接入到家庭环境的诸如音乐视频之类的音频-视频娱乐。在无线发送码流中遇到到一个问题是在广域无线网中的低带宽容量和相关的高成本问题。码流高质量视频利用高链路带宽，可以通过无线网的一种挑战。一种代替的解决方法是，在这些环境中发送码流通过广域网连接到本地无线服务器假脱机收视的视频，或和一旦已经全部或者部分被接收，通过高容量本地环路或专用无线网开始无线发送码流数据到客户机装置。

对此用于这种应用的一个实施例是本地无线音乐视频码流。用户从互联网下载音乐视频到连接无线家庭网上的本地计算机中。然后，这些音乐视频可以被发送到也是无线连接的客户机装置（例如，PDA、耐用计算机装置）。运行在本地计算机服务器上的软件管理系统管理视频库，并且响应于来自客户机装置/PDA 的客户机用户的命令控制码流处理。

服务器侧软件管理系统存在着 4 种主要部分：浏览结构产生部分；用户接口部分；码流控制部分；和网络协议部分。浏览结构产生部分产生数据结构，该结构被用于产生用于浏览本地存储视频的用户接口的数据结构。在一个实施例中，利用服务器软件，用户可以产生多个播放表，这些播放表是由用户接口部分格式化的，用于发送到客户机播放机。另外一种方案，用户可以按一种分层文件目录结构存储视频数据并且通过自动导航目录结构浏览结构部分产生浏览数据结构。用户接口部分格式化浏览数据，用于发送到客户机和从客户机接收命令，该命令被中继到码流控制部分。用户重放控制可以包括 ‘标准’ 功能，诸如重放开始、暂停、循环等。在一个实施例中，用户接口部分格式化浏览数据为 HTML，而用户重放控制为订制格式。在这个实施



例中，客户机用户接口包括两个分别的部分：HTML 浏览器处理浏览功能，而重放控制功能是由视频解码器/播放机控制。在另外的实施例中，在客户机软件中不存在分别的功能，并且视频解码器/播放机处理所有用户接口功能。在这种情况下，用户接口部分格式化浏览数据为由视频解码器/播放机直接理解的订制格式。

这种应用最适合于在家庭或公司应用中实现训练或者娱乐的目的。例如，技术员可以利用这种配置获得音频-视频材料，有关怎样修理或调整有缺陷的装置，而不必须从工作区移到在另外房间的计算机操作台。另外的应用是家庭用户在他们的院子里悠闲散步的同时，收视高质量音频-视频娱乐。返回信道允许用户从库中选择他们希望收视的音频视频内容。主要的优点是视频监视器是便携的并因此用户可以在办公室或家中自由移动。视频数据流可以象如上所述地包含多个具有交互能力的视频目标。显然，这是比电子书本和通过无线蜂窝网码流发送的现有技术的明显改善。

### 面向目标数据格式

面向目标多媒体文件格式被指定满足如下目标：

- 速度-文件被指定为按高速度修饰。
- 简单性-格式是简单的，使得快速分析和形成端口容易。另外，通过简单地添加文件就可以执行组合。
- 可扩展性-该格式是加标记格式，使得随着播放机的发展，可以定义新的包类型，同时仍然与老的播放机相兼容。
- 灵活性-从其修饰定义中存在一种数据分离，允许总的灵活性，例如改变数据速率、和编解码器中流匆忙运行。

文件按大 **endian** 字节序进行存储。使用如下数据类型：

类型	定义
<b>BYTE</b>	8 位，无符号字符
<b>WORD</b>	16 位，无符号短的
<b>DWORD</b>	32 位，无符号长的

<b>BYTE[]</b>	串, 字节[0]规定长度达 254 (255 保留)
<b>IPOINT</b>	8 位, 无符号, 12 位无符号, (x、y)
<b>DPOINT</b>	8 位无符号字符, 8 位无符号字符, (dx、dy)

文件流被分为各个包或各个数据块。每个包被包围在类似于 Quicktime 中的原子概念的容器中, 但不分层。一个容器由规定有效负荷类型的 BaseHeader 记录和某些辅助包控制信息和数据有效负荷的尺寸组成。有效负荷类型定义在流中的各种包。这种规则的一个例外是用于端对端网络链路管理的 SystemControl 包。这些包由没有有效负荷的 BaseHeader 组成。在这种情况下, 有效负荷尺寸字段被重新翻译。在流通过电路交换网络的情况下, 一个预备的附加网络容器通过提供同步和检查和被用于实现差错恢复。

在比特流中存在着 4 种主要包的类型: 数据包、定义包、控制包、和各种类型的元数据包。定义包用于传送媒体格式和被用于翻译数据包的编解码器信息。数据包传送由选择的应用程序待解码的压缩数据。因此, 适合的定义包在每个规定数据类型的任何数据包之前。定义修饰和动画参数的控制包出现在定义包之后, 但在数据包之前。

从概念上讲, 面向目标数据可以被认为是由 3 个交叉的主要数据流组成, 即定义、数据、控制流。元数据是任选的第四个流。这些 3 个主要流交互以产生最后的音频视频表示, 提供给用户。

所有文件从 SceneDefinition 块开始, 该块定义 AV 场景空间为将被修饰的任何音频或视频流或目标。元数据和目录包包含有关于由数据和定义包包含的数据的附加信息, 帮助浏览各个数据包。如果任何元数据块存在, 它们出现在紧接着 SceneDefinition 包之后。如果不存在元数据包的话, 目录包紧接着元数据包或者 SceneDefinition 包。

文件格式允许不同媒体类型的综合, 当来自远端服务器码流数据或者接入本地存储内容两者时, 支持面向目标交互。为此, 可以定义多场景并且每个可以同时含有多达 200 个分别的媒体目标。这些目标可能是单一媒体类型的, 例如视频、音频、文本或矢量图形, 或者是

这些媒体类型的组合产生的组合体。

如图 4 所示，文件结构定义各个实体的分层：文件含有一个和多个场景，每个场景可以含有一个或多个目标，并且每个目标可以含有一个或多个帧。实际上，每个场景由多个独立交叉的数据流组成，每个目标之一由多个帧组成。每个流由多个定义包之一组成，接着数据和所有都具有相同 `object_id` 号的各个控制包。

流语法

有效包类型

**BaseHeader** 按照有效负荷允许总数达 255 个不同包类型。这个部分定义对于有效包类型的包格式，列于下表中。

值	数据类型	有效负荷	注释
0	SCENEDEFN	SceneDefinition	定义场景空间性质
1	VIDEODEFN	VideoDefinition	定义视频格式/编解码器性质
2	AUDIODEFN	AudioDefinition	定义音频格式/编解码器性质
3	TEXTDEFN	TextDefinition	定义文本格式/编解码器性质
4	GRAFDEFN	GrafDefinition	定义矢量图形格式/编解码器性质
5	VIDEOKEY	VideoKey	视频关键帧数据
6	VIDEODAT	VideoData	压缩视频数据
7	AUDIODAT	AudioData	压缩音频数据
8	TEXTDAT	TextData	文本数据
9	GRAFDAT	GrafData	矢量图形数据
10	MUSIDAT	MusicData	音乐评价数据
11	OBJECTRL	ObjecControl	定义目标动画/修饰性质
12	LINKCTRL	-	用于流结束到链路管理结束
13	USERCTRL	UserControl	用户系统交互的返回信道
14	MATADATA	MetaData	含有关于 AV 场景的元数据
15	DIRECTORY	Directory	数据的目录或系统目标
16	VIDEOENH	-	保留的-视频增量数据
17	AUDIOENH	-	保留的-音频增量数据
18	VIDEOEXTN	-	用于纠错的冗余 I 帧
19	VIDEOTERP	VideoData	可抛弃的插值视频文件
20	STREAMEND	-	指示流结束和新流开始
21	MUSICDEFN	MusicDefin	定义音乐格式
22	FONTLIB	FontLibCntrol	字型库数据
23	OBJLIBCTRL	ObjectLibCntrol	目标/字型库控制
255	-	-	保留

### BaseHeader

**短的 BaseHeader 是用于比 65536 字节更短的包**

描述符	类型	注释
Type	BYTE	有效负荷包类型[0]，可以是定义，数据或控制包
Obj_id	BYTE	目标流 ID-属于哪个目标
Seq_no	WORD	帧序列号，对于每个目标的单个序列
Length	WORD	帧尺寸接着的字节{0 意味着流结束}

**长 BaseHeader 将支持从 64K 到 0xFFFFFFFF 字节的包**

描述符	类型	注释
Type	BYTE	有效负荷包类型[0]，可以是定义，数据或控制包
Obj_id	BYTE	目标流 ID-属于哪个目标
Seq_no	WORD	帧序列号，对于每个目标的单个序列
Flag	WORD	0xFFFF
Length	DWORD	帧尺寸接着字节

**系统 BaseHeader 是用于端对端网络链路管理**

描述符	类型	注释
Type	BYTE	DataType=SYSCTRL
Obj_id	BYTE	目标流 ID-属于哪个目标
Seq_no	WORD	帧序列号，对于每个目标的单个序列
Status	WORD	状态类型{ACK、NAK、CONNECT、DIS CONNECT、IDLE}+目标类型

**总尺寸是 6 到 10 字节**

**SceneDefinition**

描述符	类型	注释
<b>Magic</b>	<b>BYTE[4]</b>	<b>ASKY=0x41534B59</b> （用于格式确认）
<b>Version</b>	<b>BYTE</b>	版本 <b>0x00</b> -当前
<b>Compatible</b>	<b>BYTE</b>	版本 <b>0x00</b> -当前-最小格式可播放
<b>Width</b>	<b>WORD</b>	场景空间宽度（0=未规定）
<b>Height</b>	<b>WORD</b>	场景空间高度（0=未规定）
<b>BackFill</b>	<b>WORD</b>	保留-场景填充风格/颜色
<b>NumObjs</b>	<b>BYTE</b>	在该场景中有多少目标
<b>Mode</b>	<b>BYTE</b>	帧重放模式比特字段

总尺寸是 14 字节

**MetaData**

描述符	类型	注释
<b>NumItem</b>	<b>WORD</b>	在文件/场景中的场景/帧数（0=未规定）
<b>SceneSize</b>	<b>DWORD</b>	文件场景/目标包括的字节尺寸（0=未规定）
<b>SceneTime</b>	<b>WORD</b>	按秒的文件/场景/目标的播放时间（0=未规定/静态）
<b>BitRate</b>	<b>WORD</b>	按 k 比特/秒的文件/场景/目标的比特率
<b>MetaMask</b>	<b>DWORD</b>	规定接着是哪种任选 32 元数据标记的比特字段
<b>Title</b>	<b>BYTE[]</b>	视频文件/场景的标题-凡是您喜欢的，字节[0]=长度
<b>Creator</b>	<b>BYTE[]</b>	谁创造这个，字节[0]=长度
<b>Date</b>	<b>BYTE[8]</b>	按 ASCII 的创造日期=>DDMMYYYY
<b>Copyright</b>	<b>BYTE[]</b>	
<b>Rating</b>	<b>BYTE</b>	X、XX、XXX 等
<b>EncoderID</b>	<b>BYTE[]</b>	-
-	<b>BYTE</b>	-

**目录**

这是 WORD 或 DWORD 的阵列。尺寸是由 BaseHeader 包中的

长度字段规定的。

### **VideoDefinition**

描述符	类型	注释
Codec	BYTE	视频编解码器类型{RAW、QTREE}
Frate	BYTE	按 1/5 秒的帧速率{0=停止/暂停视频播放}
Width	WORD	视频帧宽度
Height	WORD	视频帧高度
Time	DWORD	从场景开始按 50ms 分辨率的时间标记（0=未规定）

总尺寸是 10 字节

### **AudioDefinition**

描述符	类型	注释
Codec	BYTE	音频编解码器类型{RAW、G723、ADPCM}
Format	BYTE	按比特 7-4 的音频格式、按比特 3-0 的取样速率
Fsize	WORD	每帧样值
Time	DWORD	从场景开始按 50ms 分辨率的时间标记（0=未规定）

总尺寸是 8 字节

### **TextDefinition**

描述符	类型	注释
Type	BYTE	按低半字节的类型{TEXT、HTML、等}按高半字节压缩
Fontinfo	BYTE	按低半字节字体尺寸，按高半字节字体风格
Colour	WORD	字体颜色
Backfill	WORD	背景颜色
Bounds	WORD	文本边界框（框架）X 按高字节，Y 按低字节
Xpos	WORD	如果定义，Xpos 涉及目标来源，否则涉及 0、0
Ypos	WORD	如果定义，Ypos 涉及目标来源，否则涉及 0、0
Time	DWORD	从场景开始按 50ms 分辨率的时间标志（0=未规定）

### UserControl

描述符	类型	注释
<b>Event</b>	<b>BYTE</b>	用户数据类型，例如 <b>PENDOWN</b> 、 <b>KEYEVENT</b> 、 <b>PLAYCTRL</b>
<b>Key</b>	<b>BYTE</b>	参数 1=键控代码值/开始/停止/暂停
<b>HiWord</b>	<b>WORD</b>	参数 2=X 位置
<b>LoWord</b>	<b>WORD</b>	参数 3=Y 位置
<b>Time</b>	<b>WORD</b>	Timestamp=被激活目标序列号
<b>Data</b>	<b>BYTE[]*</b>	用于从字段数据任选的字段

总尺寸是 8+字节

### ObjectControl

描述符	类型	注释
<b>ControlMask</b>	<b>BYTE</b>	定义公共目标控制的比特字段
<b>ControlObject</b>	<b>BYTE</b>	(任选)影响的目标的 ID
<b>Timer</b>	<b>WORD</b>	(任选)上半字节=定时器号，下 12 字节=100ms 步进
<b>ActionMask</b>	<b>WORD   BYTE</b>	在剩余有效负荷定义的比特字段动作
<b>Params</b>	...	由动作比特字段定义的动作参数



## 总尺寸是 16 字节

### GrafDefinition

描述符	类型	注释
<b>Xpos</b>	<b>WORD</b>	如果定义, Xpos 涉及目标来源, 否则涉及 0、0
<b>Ypos</b>	<b>WORD</b>	如果定义, Ypos 涉及目标来源, 否则涉及 0、0
<b>FrameRate</b>	<b>WORD</b>	按 8.8fps 帧延迟
<b>FrameSize</b>	<b>WORD</b>	保留, 按 2 倍 ps (1/20 像素) 的帧尺寸-用于确定比例以适合场景空间
<b>Time</b>	<b>DWORD</b>	从场景开始按 50ms 分辨率的时间标志

## 总尺寸是 12 字节

### VidieoKey、VidieoData、Audiodata、TextData、GrafData and MusicData

描述符	类型	注释
<b>Payload</b>	-	压缩的数据

### StreamEnd

描述符	类型	注释
<b>StreamObjs</b>	<b>BYTE</b>	多少目标交叉到下一个流
<b>StreamMode</b>	<b>BYTE</b>	保留
<b>StreamSize</b>	<b>DWORD</b>	按字节下一个流的长度

## 总尺寸是 6 字节

## ObjLibCtrl

描述符	类型	注释
<b>Action</b>	<b>BYTE</b>	利用这个目标进行以下操作  1.INSERT-不重写 LibID 位置  2.UPDATE-重写 LibID 位置  3.PURGE-去掉  4.QUERY-Unique_ID 目标返回 LibID/Version
<b>LibID</b>	<b>BYTE</b>	目标的索引/号在库中
<b>Version</b>	<b>BYTE</b>	这个目标版本号
<b>Persist/Expire</b>	<b>BYTE</b>	是作为垃圾收集还是放置在周围，0=会话后去掉，1-254=期满前的天，255=持续
<b>Access</b>	<b>BYTE</b>	访问控制功能  头 4 个比特：谁可以重写或去掉这个目标  1.任何会话将可以（由 LibID ）  2.系统清除/复位  3.由目标的已知的 UniqueID/LibID  4.决不/保留  比特 3： 用户是否可以转移这个目标到另一个，定向（1=是）  比特 2： 是否用户从库直接播放这个（是=1/否）  比特 1： 保留  比特 0： 保留
<b>UniqueID</b>	<b>BYTE[]</b>	这个目标的 UniqueID/label
<b>State</b>	<b>DWORD ? ? ? ?</b>	您从哪里得到它/如何、多少次跳跃、馈送时间得到，否则它消逝  1.跳跃计数  2.源（ SkyMail、 SkyFile、 SkyServer ）  3.从激活开始的时间  4.# 激活

## 语义

### BaseHeader

这是在流中所有信息包的容器。

### Type-BYTE

描述-按照上述定义规定包中有效负荷的类型

有效值：从 0 到 255，见下面有效负荷类型表

### Obj\_id- BYTE

描述-目标 ID-定义范围-这个包中属于哪个目标。

还定义在 255 个步骤中的 Z 序，朝收视者 增加

多达 4 个不同媒体类型可以共享相同 Obj\_id。

有效值：在 SceneDefinition 中定义 0- NumObjs（最大 200）个

### NumObjs

201-253：保留给系统使用

250：目标库

251：保留

252：流的目录

253：场景的目录

254：这个场景

255：这个文件

### Seq\_no-word

描述-帧序号，在一个目标内每个媒体类型的单独序列。序列号是在每个新的 SceneDefinition 包以后重新开始的。

有效值：0-0xFFFF

**Flag（任选）-WORD**

描述-用于指示长的 BaseHeader 包

有效值： 0xFFFF

**Lengh- WORD/D WORD**

用于指示按字节的有效负荷长度，（如果标志设置包尺寸=长度+0xFFFF）

有效值： 0x0001-0xFFF，如果标志设置 0x0000001-0xFFFFFFFF，0-保留用于 Endof File/Stream 0xFFFF

**Status-WORD**

使用 SysControl 数据类型标志，用于端对端链路管理。

有效值： 从 0 到 65535

值	类型	注释
0	ACK	利用规定 Obj_id 和 Seq_no 应答包
1	NAK	利用规定 Obj_id 和 Seq_no 在包上标志误差
2	CONNECT	建立客户机/服务器连接
3	DISCONNECT	断开客户机/服务器连接
4	IDLE	链路空闲
5-65535	-	保留

**SceneDefinition**

这个定义 AV 场景空间性质，将播放视频和音频目标。

**Magic-BYTE[4]**

描述-用于格式确认，

有效值： ASKY=0x41534B59

**Version-BYTE**

描述-用于系统格式确认,

有效值: 0-255 (当前=0)

**Compatible-BYTE**

描述-哪个是可以读这种格式的最小播放机

有效范围: 0-版本

**Width-WORD**

描述-像素中的 SceneSpace 宽度

有效范围: 0x0000-0xFFFF

**Height-WORD**

描述-像素中的 SceneSpace 高度

有效范围: 0x0000-0xFFFF

**BackFill- (保留) WORD**

描述-背景场景填充 (位映射、实色、倾斜)

有效范围: 0x1000-0xFFFF, 实色按 15 比特格式, 另外低序 BYTE 定义矢量目标的目标 id, 和高序 BYTE (0-15) 是倾斜填充风格表的索引。这个矢量目标定义出现在任何数据控制包之前。

**NumObjs-BYTE**

描述-在这个场景中有多少数据目标

有效范围: 0-200 (201-255 保留用于系统目标)

**Mode-BYTE**

描述-帧播放模式比特字段

比特: [7]播放状态-暂停=1, 播放=0 //连续播放或步进

比特: [6]保留作变焦-最好=1, 通常=0// 播放变焦

比特[5]保留做数据存储-活动的=1, 存储的=0 //正在发码流?

比特[4]保留发码流-可靠的=1, 最好试试=0 //是否发的码流可靠

比特[3]保留数据源-视频=1, 薄的客户机=1 //始发源

比特[2]保留交互-允许=1, 不允许=0

比特[1]保留

比特[0]库场景-该库场景 1 = 是, 0 = 否

### MetaData

这个规定与实体文件、场景或者与个别 AV 目标相关的元数据。因为这些文件可以被连接, 不保证在文件范围内元数据块在过去的场景是有效的。但是, 简单地比较文件的尺寸与在这个元数据包中的 SCENESIZE 字段可以确认这个。

在 BaseHeader 中的 OBJ\_ID 字段定义元数据包的范围。这个范围可以是整个文件( 255)、单一场景( 254)、或个别的视频目标( 0-200)。因此, 如果在文件中出现元数据包, 它们出现在紧接着 SceneDefinition 包的 flock ( packs? ) 中。

### NumItem-WORD

描述-在文件/场景中的场景/帧数。场景范围 NumItem 包含多个具有 obj\_id 的视频目标的帧。

有效范围: 0-65535 ( 0=未规定 )

### SceneSize-DWORD

描述-包含在文件/场景/目标中的字节中自包含的尺寸

有效范围: 0x0000-0xFFFFFFFF ( 0=未规定 )

### SceneTime-WORD

描述 - 按秒计的文件/场景/目标播放时间

**有效范围：0x0000-0xFFFF（0 = 未规定）**

**BitRate-WORD**

**描述-**这个文件/场景/目标的按 kbit/秒的比特速率，

**有效范围：0x0000-0xFFFF（0=未规定）**

**MetaMask-（保留的）DWORD**

**描述-**按如下次序规定任选的 32 元数据字段的比特字段，

**比特值[31]：**标题

**比特值[30]：**创建者

**比特值[29]：**创建日期

**比特值[28]：**版权

**比特值[27]：**速率

**比特值[26]：**编码器 ID

**比特值[26-27]：**保留

**标题-（任选）BYTE[]**

**描述-**多达 254 个字符的串

**创建者-（任选）BYTE[]**

**描述-**多达 254 个字符的串

**日期-（任选）BYTE[8]**

**描述-**按 ASCII=>DDMMYYYY 的创建日期

**版权-（任选）BYTE[8]**

**描述-**多达 254 个字符的串

**速率-（任选）BYTE**

## 描述-BYTE 规定 0-255

### 目录

这是对整个文件或者对场景规定目录信息。因为文件可以连接，不保证具有文件范围的元数据块在过去的场景是有效的。但是，简单地比较文件尺寸与元数据包中的 SCENESIZE 字段可以确认。

在 BaseHeader 中的 OBJ\_ID 字段定义目录包的范围。如果 OBJ\_ID 字段的值小于 200, 则该目录是对于视频数据目标的关键帧的序列号的表 (WORD)。否则, 该目录是系统目标的一个位置表。在这种情况下, 表的项目是字节 (DWORD) 中从文件开始的相关偏移 (对于场景的目录和目录) 或者对于其它系统目标的场景。表中的项目数和表的尺寸可以从 BaseHeader 包中的 LENGTH 字段计算。

类似于元数据包, 如果目录包在文件中出现, 则它们出现在紧接着 SceneDefinition 或 Metadata 包的 flocks (packs? ) 中。

### VideoDefinition

#### 编解码器-BYTE

#### 描述-压缩类型

有效值: 从 0 到 255

值	编解码器	注释
0	RAW	未压缩的, 第一字节定义色深度
1	QTREE	默认视频编解码器
2-255	-	保留

### Frate-BYTE

描述-按 1/5 秒 (即, 最大=51fps, 最小=0.2fps) 的帧播放速率

有效值: 1-255, 播放/开始播放, 如果停止 0-停止播放



**Width-WORD**

描述-视频帧中像素多宽

有效值: 0-65535

**Height-WORD**

描述-视频帧中像素多高

有效值: 0-65535

**Times-WORD**

描述-从场景开始按 50ms 分辨率的时间标记 (0=未规定)

有效值: 1-0xFFFFFFFF (0=未规定)

**AudioDefinition**

编解码器-BYTE

描述-压缩类型

有效值: 1 (0=未规定)

值	编解码器	注释
0	WAV	未压缩
1	G723	默认视频编解码器
2	IMA	交互多媒体协会 ADPCM
3-255	-	保留

**Format-BYTE**

描述-这个 BYTE 被分裂为 2 个独立定义的单独字段。上 4 个比特定义音频格式 (格式 >>4) 而下 4 个比特分别定义样值速率 (格式 &0xF)。

低位 4 个比特, 值: 从 0 到 15, 取样速率

值	取样速率	注释
---	------	----

0	0	0-停止播放
1	5.5kHz	5.5kHz 非常低的速率取样，如果停止开始播放
2	8kHz	标准 8000Hz 取样，如果停止开始播放
3	11kHz	标准 11025Hz 取样，如果停止开始播放
4	16kHz	2×8000Hz 取样，如果停止开始播放
5	22kHz	标准 22050Hz 取样，如果停止开始播放
6	32kHz	4×8000Hz 取样，如果停止开始播放
7	44kHz	标准 44100Hz 取样，如果停止开始播放
8-15		保留

比特 4-5，值：从 0 到 3，格式

值	格式	注释
0	MONO8	单声道，每样值 8 比特
1	MONO6	单声道，每样值 16 比特
2	STEREO8	立体声，每样值 8 比特
3	STEREO16	立体声，每样值 16 比特

高位 2 个比特（6-7），值：从 0 到 3，特殊的

编解码器	注释
WAV	保留（未使用）
G.723	保留（未使用）
IMA	每样值比特（值+2）

**Fsize-WORD**

描述-每帧样值

有效值：0-65535

## Times-WORD

描述-从场景开始按 50ms 分辨率的时间标记（0=未规定）

有效值：1-0xFFFFFFFF（0=未规定）

## Text Definition

我们需要包括写方向，它可以是 LRTB、RLTB、TBRL、或 TBLR。这可以利用文本的主体中的特殊字母代码指示方向，例如我们可以利用 DC1-DC4（ASCII 装置控制代码 17-20）完成这个任务。我们还需要具有在位映射字型的开始下载的字型表。取决于播放机运行平台，修饰可以忽略位映射字型或者试图使用位映射字型修饰文本。如果没有位映射字型表或者如果被播放机忽略，修饰系统将自动使用操作系统文本输出功能修饰文本。

## Type-BYTE

描述-定义多少文本数据在低半字节（Type & 0x0F）翻译和在高半字节（Type>>4）的压缩方法

低半字节，值：从 0 到 15，类型-翻译

值	类型	注释
0	PLAIN	无格式文本 1-不翻译
1	TABLE	保留-表数据
2	FORM	用于用户输入的形式/文本字段
3	WML	保留的 WAP-WML
4	HTML	保留 HTML
5-15	-	保留

**高位 4 比特，值：从 0 到 15，压缩方法**

值	类型	注释
0	NONE	无压缩 8 比特 ASCII 码
1	TEXT7	保留的 7 比特字符码
2	HUFF4	保留的-4 比特 Huffman 编码 ASCII
3	HUFF8	保留的-8 比特 Huffman 编码 ASCII
4	LZW	保留的-Lempel-Zev-Welsh 编码 ASCII
5	ARITH	保留的-算法编码 ASCII
6-15	-	保留的

### FontInfo-BYTE

描述-在低半字节的尺寸 (FontInfo & 0x0F )，在高半字节的风格 (FontInfo>>4)

如果类型是 WML 或 HTML，这个字段忽略。

低位 4 比特，值：0-15    FontSize

高位 4 比特，值：0-15    FontSyle

### Colour-WORD

描述-文本表面色

有效值：0x0000-0xEFFF，15 比特 RGB ( R5、G5、B5 )  
0x8000-0x80FF 中的色，色作为 VideoData LUT (0x80FF=透明) 中的索引

0x8100-0xFFFF 保留

### BackFill-WORD

描述-背景色

有效值：0x0000-0xEFFF，15 比特 RGB ( R5、G5、B5 )  
0x8000-0x80FF 中的色，色作为 VideoData LUT (0x80FF=透明) 中

## 的索引

### **0x8100-0xFFFF 保留**

#### **Bounds-WORD**

描述-字符单元中的文本边界框（框架），宽度按 LoByte ( Bound & 0x0F ), 和高度按 HiByte( Bound >>4 ). 文本将利用宽度限制( wrap ) 和高度剪切。

有效值：宽度=1-255，高度=1-255，宽度=0-不执行限制，高度=0-不执行剪切

#### **Xpos-WORD**

描述-如果定义， pos 涉及目标来源，否则涉及 0, 0

有效值： 0x0000-0xFFFF

#### **Ypos-WORD**

描述-如果定义， pos 涉及目标来源，否则涉及 0, 0

有效值： 0x0000-0xFFFF

注意：在 0x80F0-0x80FF 范围内的色是加入 VideoData LUT 的无效色索引，因为它们仅支持到 240 色。因此，它们按照如下表翻译。根据这个表按照最好的可能，这些色应当被映射到特定装置/OS 系统色。在标准 Palm OS UI 中，仅使用 8 色并且这些色的某些色类似于其它平台，但不相同，这是表示不保持固定状态。丢失的 8 色将必须通过应用程序进行设置。

#### **GrafDefinition**

这个包含有基本动画参数。实际图形目标定义被包含在 GrafData 包中，并且动画控制包含在 objControl 包。

**Xpos-WORD**

描述- 如果定义, Xpos 涉及目标来源, 否则涉及 0, 0

有效值:

**Ypos-WORD**

描述-如果定义, Xpos 涉及目标来源, 否则涉及 0, 0

有效值:

**FrameRate-WORD**

描述-按 8.8fps 的帧延迟

有效值:

**FrameSize-WORD**

描述-按两倍 ps (1/20pel) 的帧尺寸-用于确定比例以适应场景空

间

有效值:

**FrameCount-WORD**

描述-在这个动画中有多少帧

有效值:

**Time-DWORD**

描述-从场景开始按 50ms 分辨率进行时间标记

有效值:

**VideoKey、VideoData、VideoTrp 和 AudioData**

这些包含有编解码器特殊的压缩数据。

缓冲器尺寸将从在 VideoDefn 和 AudioDefn 包传送的信息中确定。在远处的 TypeTag VideoKey 包类似于 VideoData 包, 不同仅在

于它们能够编码透明区- VideoKey 帧没有透明区。在类型定义上的区别使得在文件分析级别上关键帧是可视的，容易浏览。VideoKey 包是 VideoData 包序列的整体部件；一般它们散布在 VideoData 包序列之中，作为相同包序列的部分。VideoTrp 包代表对于视频流非实质性的帧，因此它们可以被 Syk 解码引擎丢弃。

### TextData

文本数据包包含用于将被修饰的文本的 ASCII 字符码。无论哪种衬线（serif）系统字型是可用的，一个客户机装置将被用于修饰这些字型。因为成比例的字型要求附加的处理进行修饰，衬线字型将是使用的。在特殊衬线系统的字型风格是不可以使用的情况下，则将使用最接近匹配的可用字型。

无格式文本被直接修饰，不用任何翻译。规定如下的非 LF（新行）字符的白空格字符和用于表格与形式的空格与其它特殊码都被忽略并且跳过。所有文本在场景边界进行剪裁。

边界框定义文本如何进行限制功能。文本将利用宽度进行限制和如果超高进行剪裁。如果边界宽度是 0，则不进行限制。如果高度是 0，则不进行剪裁。

表数据类似于具有 LF 例外的无格式文本进行处理，被用于行的结束，并且 CR 字符被用于表示列中断。

WML 和 HTML 根据它们各自的标准进行翻译，并且在这些格式中规定的字型风格被忽略。在 WML 和 HTML 中不支持图象。

为了获得码流文本数据，新的 TextData 被发送，更新相关的目标。另外在正常文本动画中，可以利用 ObjectControl 包定义 TextData 的修饰。

### GrafData

这个包含有用于图形动画的所有图形形状和风格定义。这是非常简单的动画数据类型。每个形状是由路径、某些属性和画法风格定

义的。一个图形目标可以包含任何一个 **GraphData** 包中的路径阵列。这个图形目标的动画可以通过清除或者替换个别的形状记录阵列整体在下一帧中发生，利用执行 **CLEAR** 和 **SKIP** 路径类型，还可以执行把新的记录增加到阵列。

### **GrafData Packet**

描述	类型	注释
<b>NumSapes</b>	<b>BYTE</b>	形状的数量记录到下面
<b>Primitives</b>	<b>SHAPERecord[]</b>	形状定义的阵列

### **ShapeRecord**

描述	类型	注释
<b>Path</b>	<b>BYTE</b>	设置形状+DELETE 操作的路径
<b>Style</b>	<b>BYTE</b>	定义路径如何被翻译和修饰
<b>Offset</b>	<b>IPOINT</b>	
<b>Vertices</b>	<b>DPOINT[]</b>	在路径低半字节规定的阵列长度
<b>FillColour</b>	<b>WORD[]</b>	取决于填充风格和#顶端的项目数
<b>LineColour</b>	<b>WORD</b>	由风格字段确定的任选字段

### **Path-BYTE**

描述-设置在高半字节中的形状路径和低半字节中的#顶端

低位 4 比特值：0-15，多路径中的顶端数

高位 4 比特值：ENUMERATED: 0-15 定义路径形状



值	路径	注释
0	<b>CLEAR</b>	从阵列删除 SHAPERECORD
1	<b>SKIP</b>	跳阵列中的这个 SHAPERECORD
2	<b>RECT</b>	描述-左上角、右下角，有效值：（0..4096、0..4096）、[0..255、0..255]
3	<b>POLY</b>	描述-各#点、初始 xy 值、相关 pt 座标阵列 有效值：0..255、（0..4096、0..4096）、[0..255、0..255]
4	<b>ELLIPSE</b>	描述-中心座标、主轴半径、次轴半径 有效值：（0..4096、0..4096）、0..255、0..255
5-15		保留

## Style-BYTE

描述-定义路径如何翻译

低位 4 比特值：0-15 行密度

高位 4 个比特：BITFIELD：路径修饰参数。所有默认没有表示出形状，使得按照不可见热区进行操作。

比特[4]：CLOSED-如果设置该比特，则路径关闭

比特[5]：FILLFLAT-默认不填充-如果填充，则不操作

比特[6]：FILLSHADE-默认不填充-如果填充，则不操作

比特[7]：LINECOLOR-默认不概述

## UserControl

用于控制用户系统和用户目标交互事件。用作反向信道返回用户交互到服务器，影响服务器侧的控制。但是，如果文件不是发码流进行的，这些用户交互由客户机处理。在每个包中对于用户目标控制可以定义多个动作。在这个版本中定义如下动作。用户目标交互不需要规定，除了通知服务器交互已经发生，因为服务器知道哪些动作是有效的。

用户-系统交互	用户-目标交互
笔事件（上、下、移动、双击鼠标）	设置 <b>2D</b> 位置、可见性（自己、其它）
键盘事件	播放/暂停系统控制
播放控制（播放、暂停、帧前进、停止）	超链路-到#（场景、帧、标注、URL）
返回形成数据	超链路-到下一个/前一个，（场景、帧）
	超链路-替换目标（自己、其它）
	超链路-服务器定义

用户目标交互取决于，当由用户点击时每个目标的哪个被定义。通过 **ObjectControl** 消息的媒体播放机可以知道这些动作。如果不知道，它们可以被传送到在线服务器用于处理。利用用户-目标交互在 **BaseHeader** **obj\_id** 字段指示相关目标的识别。这应用 **OBJCTRL** 和 **FORMDATA** 事件类型。对于用户-系统交互，**obj\_id** 字段的值是 255。在 **UserControl** 包中的事件类型规定按键、**HiWord** 和 **LoWord** 数据字段的翻译。

### **Event-BYTE**

描述-用户事件类型

有效值：从 0 到 255

值	事件类型	注释
0	PENDOWN	用户把笔放在触摸屏上
1	PENUP	用户把笔从触摸屏抬起
2	PENMOVE	用户用笔划过触摸屏
3	PENDBLCLK	用户用笔双击触摸幕
4	KEYDOWN	用户按键
5	KEYUP	用户按键
6	PLAYCTRL	用户激活播放/暂停/停止控制按钮
7	OBJCTRL	用户点击/激活 AV 目标
8	FORMDATA	用户从数据返回
9-255	-	保留

**按键、HiWord 和 LoWord-BYTE、WORD、WORD**  
**描述-用于不同事件类型的参数数据**  
**有效值：这些字段翻译如下**

事件	按键	HiWord	LoWord
PENDOWN	如果按键按下的键码	X 位置	Y 位置
PENUP	如果按键按下的键码	X 位置	Y 位置
PENMOVE	如果按键按下的键码	X 位置	Y 位置
PENDBLCLK	如果按键按下的键码	X 位置	Y 位置
KEYDOWN	键码	统一代码的键码	第二键按下
KEYUP	键码	统一代码的键码	第二键按下
PLAYCTRL	停止=0、开始=1、暂停=2	保留	保留
OBJCTRL	笔事件 ID	如果按键按下的键码	保留
FORMDATA	保留	数据字段的长度	保留

**Time-WORD**

**描述-用户事件的时间=被激活目标的序列号**

**有效值： 0-0xFFFF**

### **Data- ( RESERVED-OPTIONAL )**

**描述-来自形式目标的文本串**

**有效值： 0...65535 字节长度**

**注示：**在播放业已暂停但重复按暂停的 **PLAYCTRL** 事件的情况下，应当从服务器调用帧前置响应。停止应当将播放复位，到文件/流的开始。

## **ObjectControl**

**ObjectControl** 包被用于定义目标-场景和系统场景的交互。它们还具体定义如何修饰目标和如何播放场景。一个新的 **OBJCTRL** 包被用于每个帧，对个别目标布局定座标。在每个包中对每个包中可以定义多个动作。在这个版本中定义如下动作

目标系统动作	系统-场景动作
设置 2D/3D 位置	到#（场景、帧、标记、URL）
设置 3D 旋转	到下一个、前一个（场景、帧）
设置比例/尺寸系数	播放/暂停
设置可见度	静音
设置标记/标题（作为工具提示使用）	如果（场景、帧、目标），则操作（动作）
设置背景色（0 = 透明的）	
设置中间值（用于动画）	
开始/结束/持续/重复（循环）	
暗示	

### **• Control-BYTE**

◦ **描述-比特字段-控制屏蔽**定义对目标级别和系统级别操作的控制。接着 **ControlMask** 是指示受影响目标的目标 id 的任选参数。如

果没有受影响的规定目标 ID，则受影响 id 是基本首端的目标 id。接着 ControlMask 的 ActionMask（目标或系统范围）的类型是由受影响的目标 id 确定的。

- 比特：[7]CONDITION-执行这些动作需要哪个
- 比特：[6]BACKCOLR-设置目标背景色
- 比特：[5]PROTECT-限制用户对场景目标的修改
- 比特：[4]JUMPTO-用另外的替换一个目标的源流
- 比特：[3]HYPERLINK-设置超链路对象
- 比特：[2]OTHER-受影响目标的目标 id 将接着（255=系统）

- 比特：[1]SETTIMER-设置定时器和开始下计数
- 比特：[0]EXTEND-保留用于将来扩展

- ControlObject-BYTE（任选）

- 描述：受影响目标的 ID。如果 ControlMask 的比特被设置，则被包括

- 有效值：0-255

- Timer-WORD（任选）

- 描述：上半字节=定时器号，下 12 比特=时间设置
  - 上半字节，有效值：0-15 对于这个目标的定时器号
  - 下 12 比特有效值范围：0-4096 时间设置，按 100ms 步进

- ActionMask[OBJECT scope]-WORD

- 描述-比特字段-定义在这个记录和如下参数中规定哪个动作。对其它系统范围的一个目标存在两个版本。这个字段定义应用到媒体目标上的动作。

- 有效值：对在 ActionMask16 中 16 比特之一的每个目标识别将要采取的一个动作。如果设置一个比特，则附加相关参数值接着这个字段。

- 比特：[15]BEHAVIOR -指示这个动作和条件保持该目标，即使该动作已经执行后

- 比特: [14]ANIMATE-定义路径的多控制点将如下
- 比特: [13]MOVETO-设置动作位置
- 比特: [12]ZORDER-设置深度
- 比特: [11]ROTATE-3D 取向
- 比特: [10]ALPHA-透明度
- 比特: [9]SCALE-比例/尺寸
- 比特: [8]VOLUME-设置响度
- 比特: [7]FORECOLR-设置/改变前景色
- 比特: [6]CTRLLOOP-重复下面的#动作 (如果另设置  
ENDLOOP)

- 比特: [5] ENDLOOP-如果循环控制/动画, 则中断
- 比特: [4]BUTTON-用按钮定义 penDown 图象
- 比特: [3]COPYFRAME-从目标复制帧到这个目标 (检查框)
- 比特: [2]CLEAR\_WAITING\_ACTIONS-清等待动作
- 比特: [1]OBJECT\_MAPPING-规定各个流之间的目标映射
- 比特: [0]ACTIONEXTEND-扩展动作屏蔽跟随

• ActionExtend[OBJECT 范围]-WORD

- 描述-比特字段-保留

• ActionMask[SYSTEM 范围]-BYTE

◦ 描述-比特字段-定义哪个动作被规定在这个记录和接着的参数中。对其它系统范围的一个目标存在两个版本。这个字段定义具有场景宽度范围的动作。

◦ 有效值: 对在 ActionMask 中 16 比特之一的每个系统识别将要采取的一个动作。如果设置一个比特, 则附加相关参数值接着这个字段。

- 比特: [7]PAUSEPLAY-是否播放暂停
- 比特: [6]SNDMUTE-如果发声则静音, 如果静音则有声
- 比特: [5]SETFLAG-设置用户可指定的系统标志值
- 比特: [4]MAKECALL-改变/打开物理信道

- 比特: [3]SENDDTMF-在语音呼叫中发 DTMF 音调
- 比特: [2-0]-保留

- Params-BYTE 阵列

- 描述-比特阵列。在上述比特字段中定义的主要动作使用附加参数。被设置的由比特字段值指示的所用参数被按与用于屏蔽的从上（15）到下（0）的比特字段相同次序规定在这里。ActionMask 则 [Object/System]Mask（除了在两个之间业已规定的受影响的 object id 外）。这些参数可以包括任选字段，这些在下表中表示为黄色的行。

- CONDITION 比特-由链接在一起的一个或多个状态记录组成，每个记录还可以在其后具有任选的帧号字段。每个记录中的各个条件被逻辑“与”。为了更大的灵活性，附加记录可以通过比特 0 进行链接，产生逻辑“或”条件。除此以外，对于产生多个用于每个目标的条件控制路径的任何一个目标，多个不同定义记录可以存在。

参数	类型	注释
<b>State</b>	<b>WORD</b>	执行这些动作需要哪个比特，比特字段（被逻辑“与”） <ul style="list-style-type: none"> <li>•比特：[15]播放//连续播放</li> <li>•比特：[14]暂停//播放暂停</li> <li>•比特：[13]码流//从远端服务器发码流</li> <li>•比特：[12]存储//从本地存储器播放</li> <li>•比特：[11]缓冲//是否目标帧#被缓冲（如果存储是真）</li> <li>•比特：[10]重叠 //需要拖下哪个目标？</li> <li>•比特：[9]事件//将发生哪个用户事件？</li> <li>•比特：[8]等待//等待条件变真？</li> <li>•比特：[7]用户标志//测试接着的用户标志</li> <li>•比特：[6]时间达到//定时器已期满</li> <li>•比特：[5-1]保留</li> <li>•比特：[0]“或”状态//“或”状态条件记录跟随</li> </ul>
<b>Frame</b>	<b>WORD</b>	（任选）比特 11 条件的帧数
<b>Object</b>	<b>BYTE</b>	（任选）比特 10 条件的目标 ID，可以用于不可见目标
<b>Event</b>	<b>WORD</b>	高 BYTE：来自用户控制包的事件字段 低 BYTE：来自用户控制包的按键字段 0xFF 忽略按键，0x00 没有按键
<b>User</b>	<b>DWOR</b>	高 WORD：屏蔽指示检查哪些标志
<b>flags</b>	<b>D</b>	低 WORD：屏蔽指示用户标志值（设置或未设置）
<b>TimeUp</b>	<b>BYTE</b>	高半字节：保留 低半字节：定时器 id 号（0-15）
<b>State</b>	<b>WORD</b>	与前面状态字段相同的比特字段，但与它逻辑“或”
<b>...</b>	<b>WORD</b>	...

° ANIMATE 比特设置-如果设置动画比特，则动画参数接着规定



**时间和动画插值。动画比特还影响：存在于这个控制的 MOVETO、ZORDER、ROTATE、ALPHA、SCALE、和 VOLUME 参数。对于每个参数将有多值，每个控制点一个值。**

参数	类型	注释
AnimCtr1	BYTE	高半字节：控制点数-1 低半字节：路径控制 •比特：[3]循环动画 •比特：[2]保留 •比特：[1..0]可列举的，路径类型-{0: 线性，1: 二次方的，2: 立方的}
开始时间	WORD	动画开始时间，从场景开始或按 50ms 步进条件
持续时间	WORD[]	按 50ms 增量持续期阵列，长度=控制点-1

### ° MOVETO 比特设置

参数	类型	注释
Xpos	WORD	相对当前位置，X 位置移动到
Ypos	WORD	相对当前位置，Y 位置移动到

### ° ZORDER 比特设置

参数	类型	注释
Depth	WORD	远离收视者的深度增加，保留值 0,256,512,768 等

### ° ROTATE 比特设置

参数	类型	注释
<b>Xrot</b>	<b>BYTE</b>	<b>X 轴旋转，绝对值按度*255/360</b>
<b>Yrot</b>	<b>BYTE</b>	<b>Y 轴旋转，绝对值按度*255/360</b>
<b>Zrot</b>	<b>BYTE</b>	<b>Z 轴旋转，绝对值按度*255/360</b>

### ° ALPHA 比特设置

参数	类型	注释
<b>alpha</b>	<b>BYTE</b>	透明度，0=透明、255=全不透明

### ° SCALE 比特设置

参数	类型	注释
<b>scale</b>	<b>WORD</b>	按 8.8 固定内部格式的寸/比例

### ° VOLUME 比特设置

参数	类型	注释
<b>vol</b>	<b>BYTE</b>	音量，0=最小，255=最大

### ° BACKCOLR 比特设置

参数	类型	注释
<b>fillcolr</b>	<b>WORD</b>	与 SceneDefinition 背景色相同格式（0=透明）

### ° PROTECT 比特设置

参数	类型	注释
<b>Protect</b>	<b>BYTE</b>	限制用户修改场景目标比特字段，比特设置=不允许 •比特：[7]移动 //禁止移动目标 •比特：[6] $\alpha$ //禁止改变 $\alpha$ 值 •比特：[5]深度 //禁止改变深度值 •比特：[4]点击 //禁止点击行为 •比特：[3]拖 //禁止拖目标 •比特：[2...0] //保留

### ° CTRLLOOP 比特设置

参数	类型	注释
<b>Repeat</b>	<b>BYTE</b>	重复这个目标的下面的#动作-点击目标中断循环

### ° SETFLAG 比特设置

参数	类型	注释
<b>flag</b>	<b>BYTE</b>	上半字节=标志号，下半字节如果是真，设置标志否则复位标志

### ° HYPERLINK 比特设置

参数	类型	注释
<b>hLink</b>	<b>BYTE[]</b>	对点击设置超链路目标 URL

### ° JUMPTO 比特设置

参数	类型	注释
Scene	BYTE	到场景#, 如果值=0xFF, 到超链路 (250=库)
Stream	BYTE	[任选]码流#, 如果值=0 则读任选目标 id
Object	BYTE	[任选]目标 id#

### ° BUTTON 比特设置

参数	类型	注释
Scene	BYTE	场景# (250=库)
Stream	BYTE	码流#, 如果值=0 则读任选目标 id
Object	BYTE	[任选]目标 id#

### ° COPYFRAME 比特设置

参数	类型	注释
Object	BYTE	将从具有该 id 的目标复制帧

° OBJECTMAPPING 比特设置-当一个目标跳到另外的流时, 该流可以使用与当前场景不同目标 id。因此目标映射规定为在含有 JUMPTO 命令的相同包中。

参数	类型	注释
Object	BYTE	待映射的目标数
Mapping	WORD[]	字的阵列, 长度=各个目标 高 BYTE: 用于我们跳到的流的目标 id 低 BYTE: 将映射到新目标的当前场景的目标 id

### ° MAKECALL 比特设置

参数	类型	注释
channel	DWORD	新信道的电话号码

### ° SENDDTMF 比特设置

参数	类型	注释
DTMF	BYTE[]	待发送到信道的 DTMF 串

注示:

- 没有用于 PAUSEPLAY 和 SNDMUTE 动作的参数,因为它们 是二进制标志。
- 通过在初始透明设置的额外图象目标可以产生按钮状态。当用 户点击按钮目标时,则这个目标被不可见目标替换,即利用按钮行为 字段设置可见的目标并且当笔抬起时回到原来的状态。

## ObjLibControl

ObjLibCtrl 包被用于控制播放机维持的持久本地目标库。在一个 场景中,在一个场景中,本地目标库可以被认为存储资源。总数 200 个用户目标和 55 个系统目标可以被存储在每个库中。在重放期间利用 object\_id=250 目标库可以对于场景直接进行寻址。目标库是非常强有 力的并且不像字型库,支持持久和自动收集垃圾两者。

通过组合具有设置在 Mode 比特字段[比特 0]中的 ObjLibrary 比 特的 ObjLibCtrl 包和 SceneDefn 包,将各个目标插入目标库。在 SceneDefn 包中设置这个比特告诉播放机,接着这个比特的数据将不 直接进行播出,而是为了增殖目标库。用于目标库的实际目标数据不 以任何特殊方式打包,而仍然由定义包和数据包组成。差别是现在存 在用于每个目标的相关的 ObjLibCtrl 包,指令播放机在场景中用目标 数据作哪些事情。每个 ObjLibCtrl 包含有对基本首端中具有相同

**object\_id** 的目标的管理信息。一种 **ObjLibCtrl** 包的特殊情况是在基本首端中的 **object\_id** 设置为 250。这些被用于传送库系统管理命令到播放机。

描述在这里的本发明可以根据提供的规范技术应用通用数字计算机或者本说明发描述编程微处理器方便地实施，以及对于计算机领域的技术人员将是显而易见的。由熟练的编程员根据本发明提供的技术可以制备适当的软件编码，这对本专业的技术人员也是显而易见的。本发明还可以通过制备专用集成电路或者通过互联常规部件电路的适当网络来实施，这对本专业的技术人员也是显而易见的。应当注意到，本发明不仅包括披露在这里的编码处理和系统，而且还包括对应的解码系统和处理，后者可以操作解码编码的比特流或由编码产生的文件实现的，而后者基本上是避开编码的某些步骤的编码的相反次序的操作。

本发明包括计算机程序产品或制造的物品，它们是包含指令的存储介质，这些指令可以用于编程计算机或者计算装置，以执行本发明的处理过程。存储介质可以包括，但不限于任何类型的盘，包括：软盘、光盘、**CD-ROM**、和磁光盘、**ROM**、**RAM**、**EPROM**、**EEPROM**、磁或光卡、或者适于存储电子指令的任何类型的媒介。本发明还包括由本发明的编码处理产生的数据或信号。这种数据或信号可以是以电磁波的形式或者存储在适于的存储媒介中的。

在不脱离这里所描述的本发明的精神和范围的情况下，本专业的技术人员作出许多修改将是显而易见的。

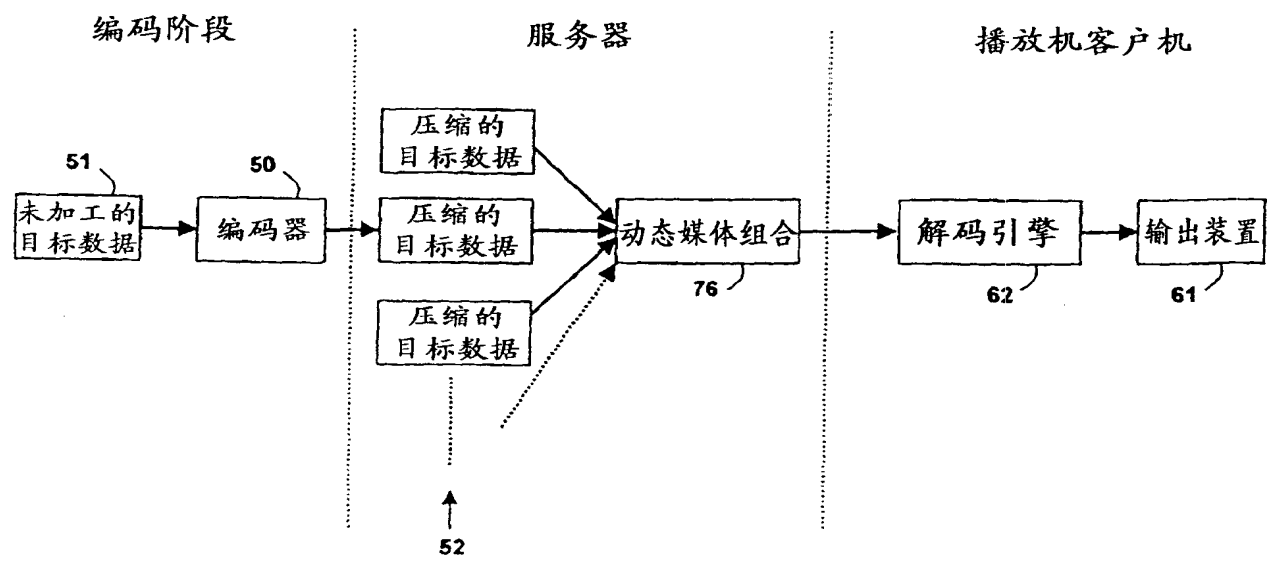


图 1

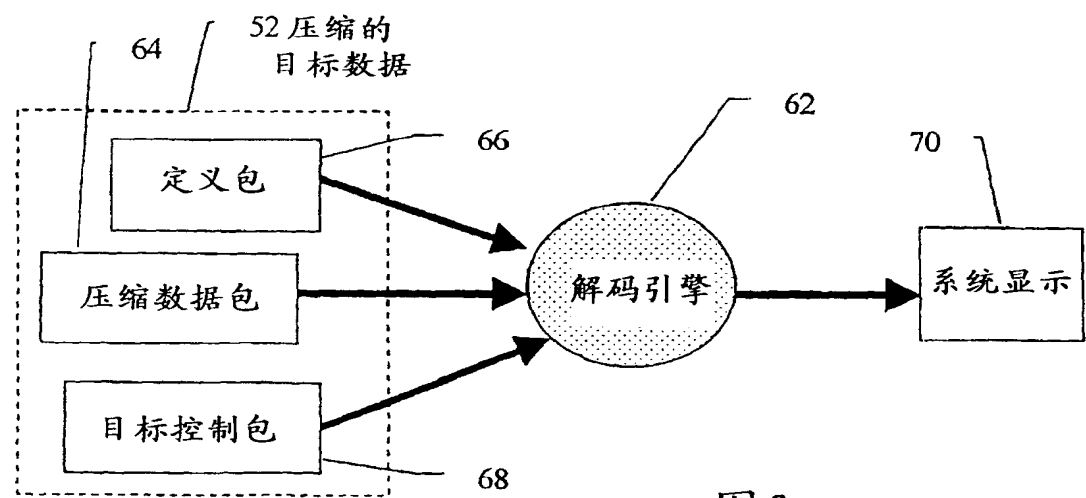


图 2

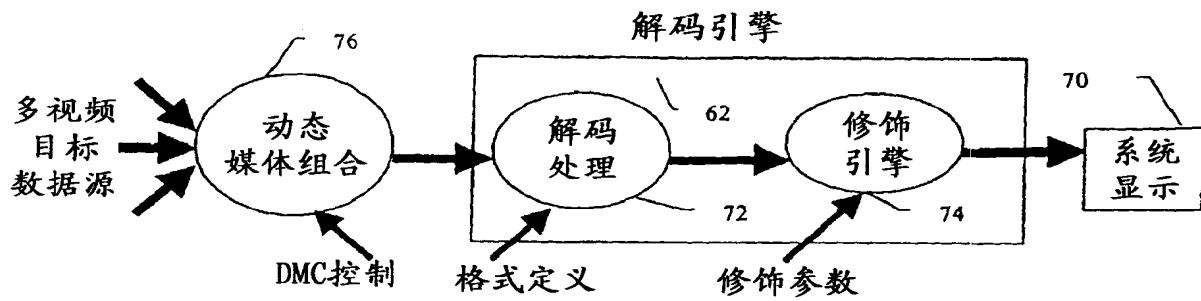


图3

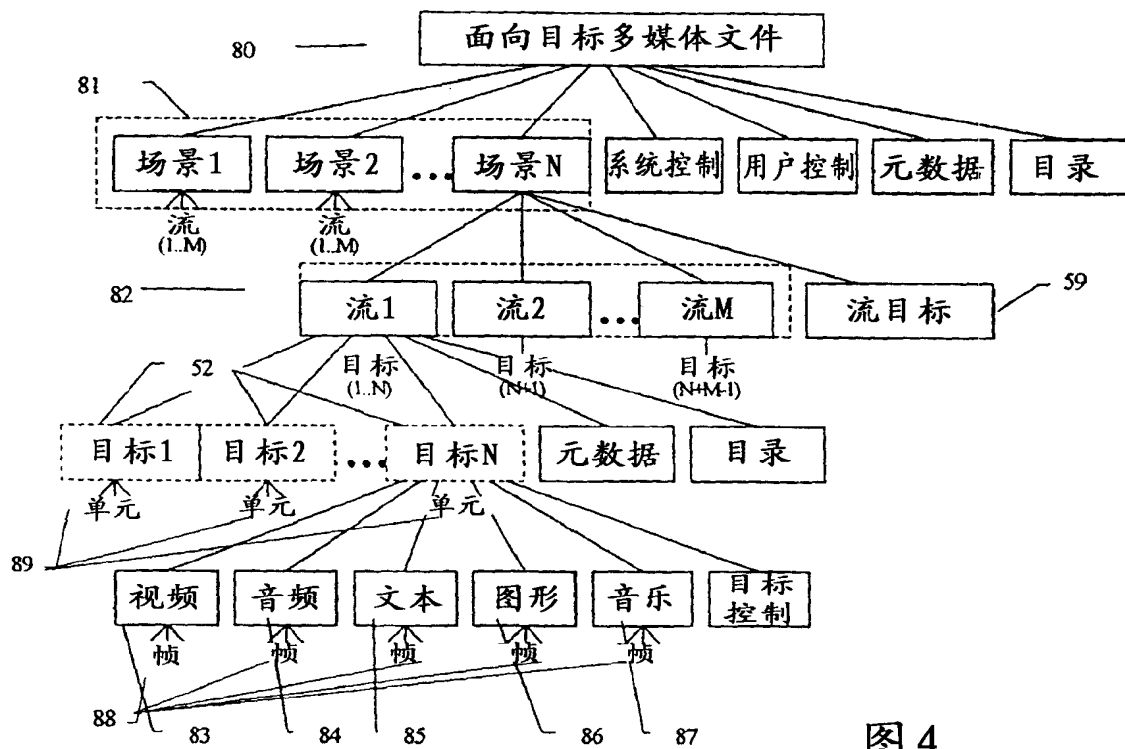


图4



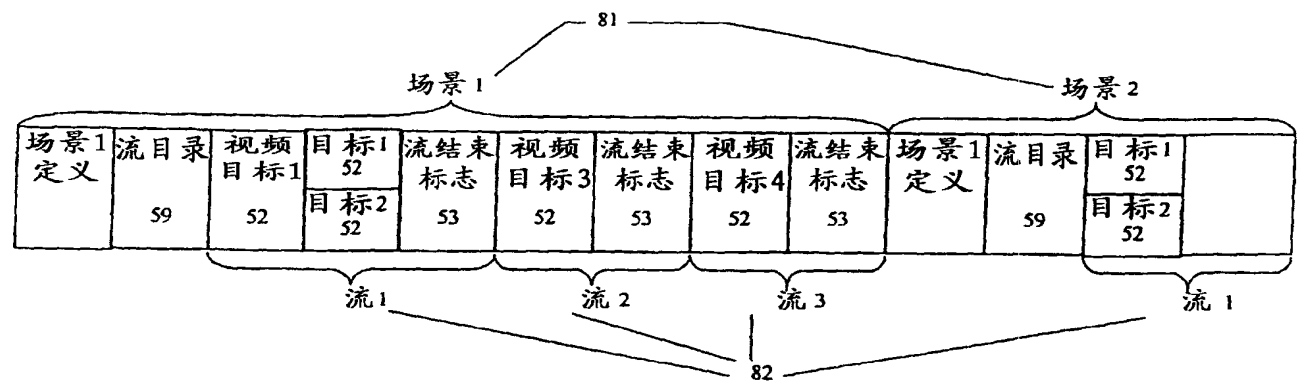


图 5

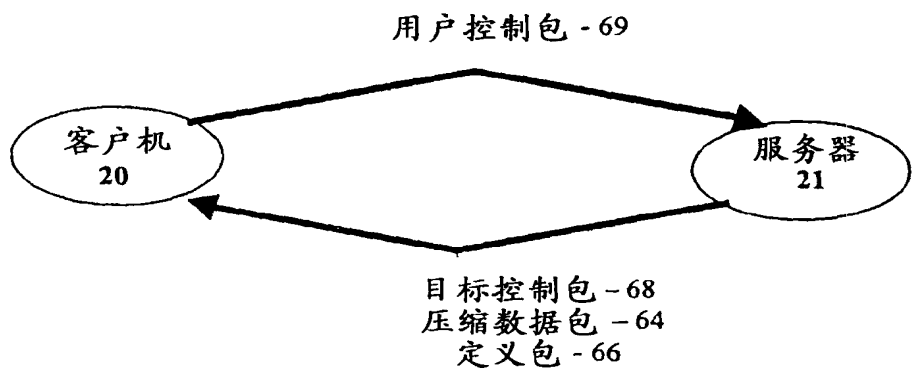


图 6

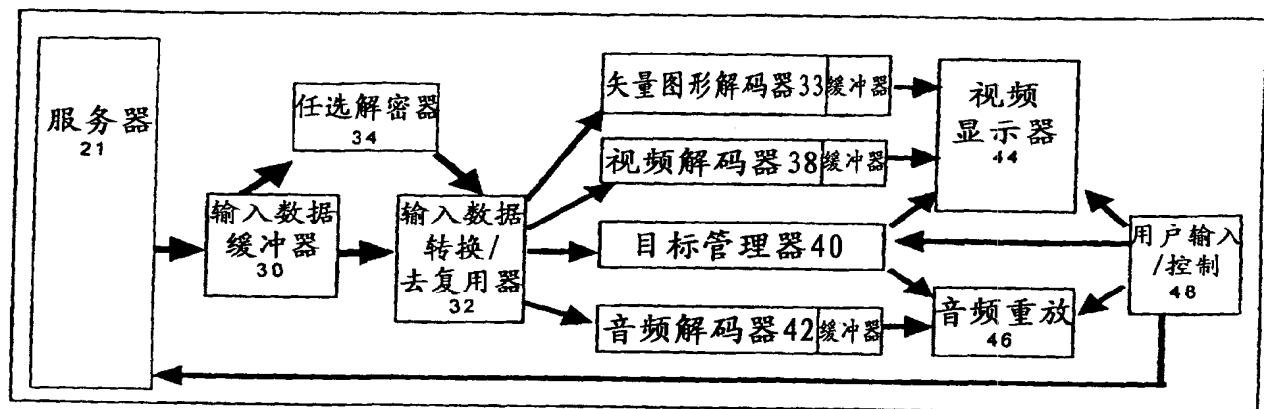


图 7

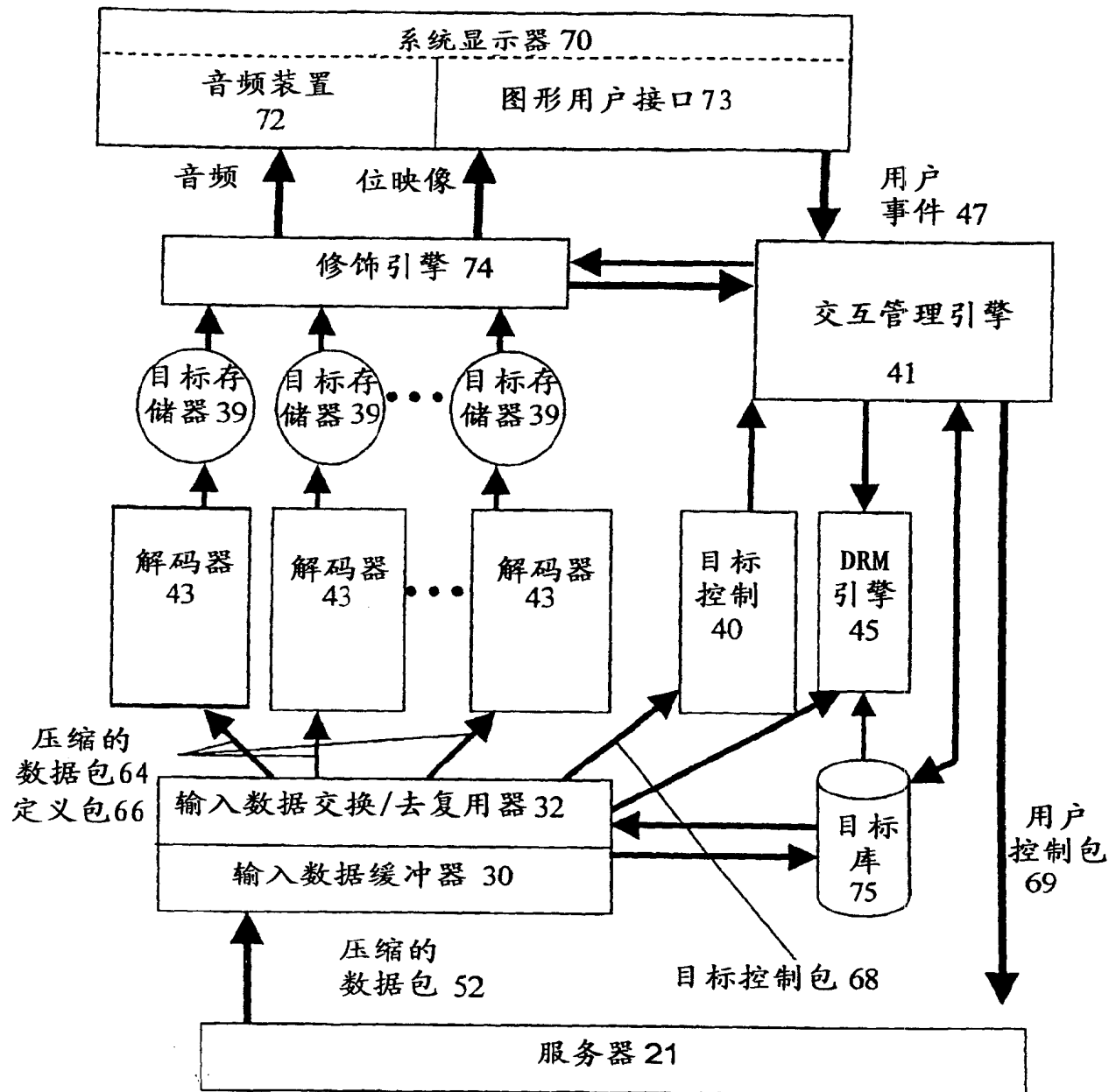


图 8

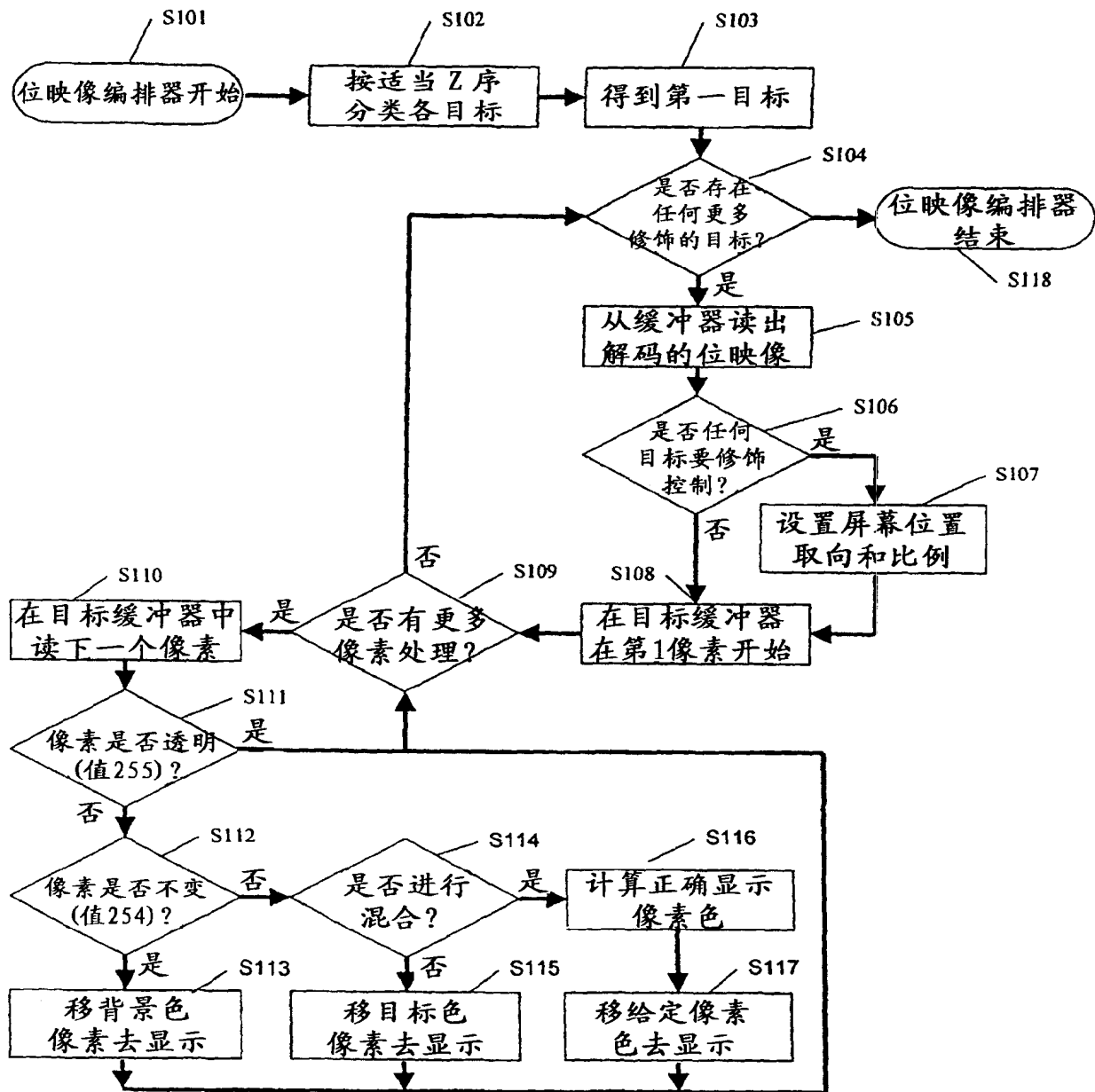


图9

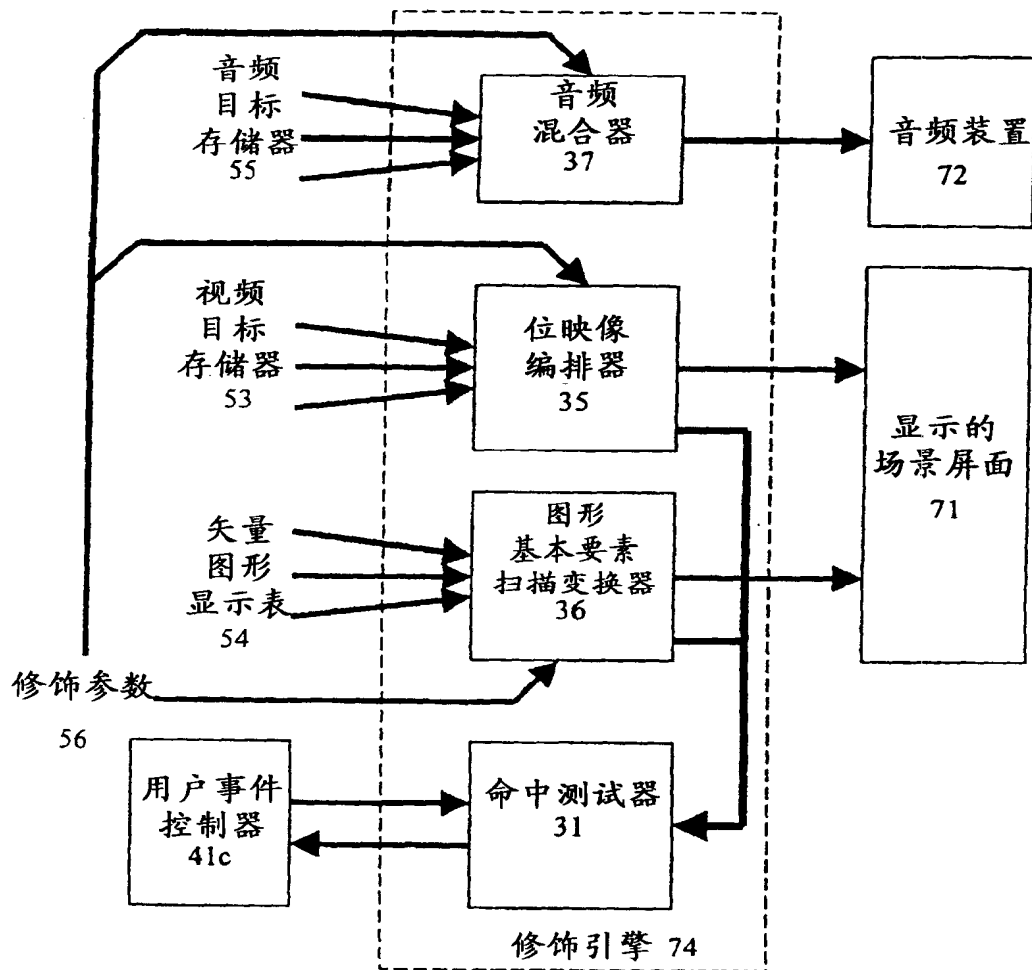


图 10

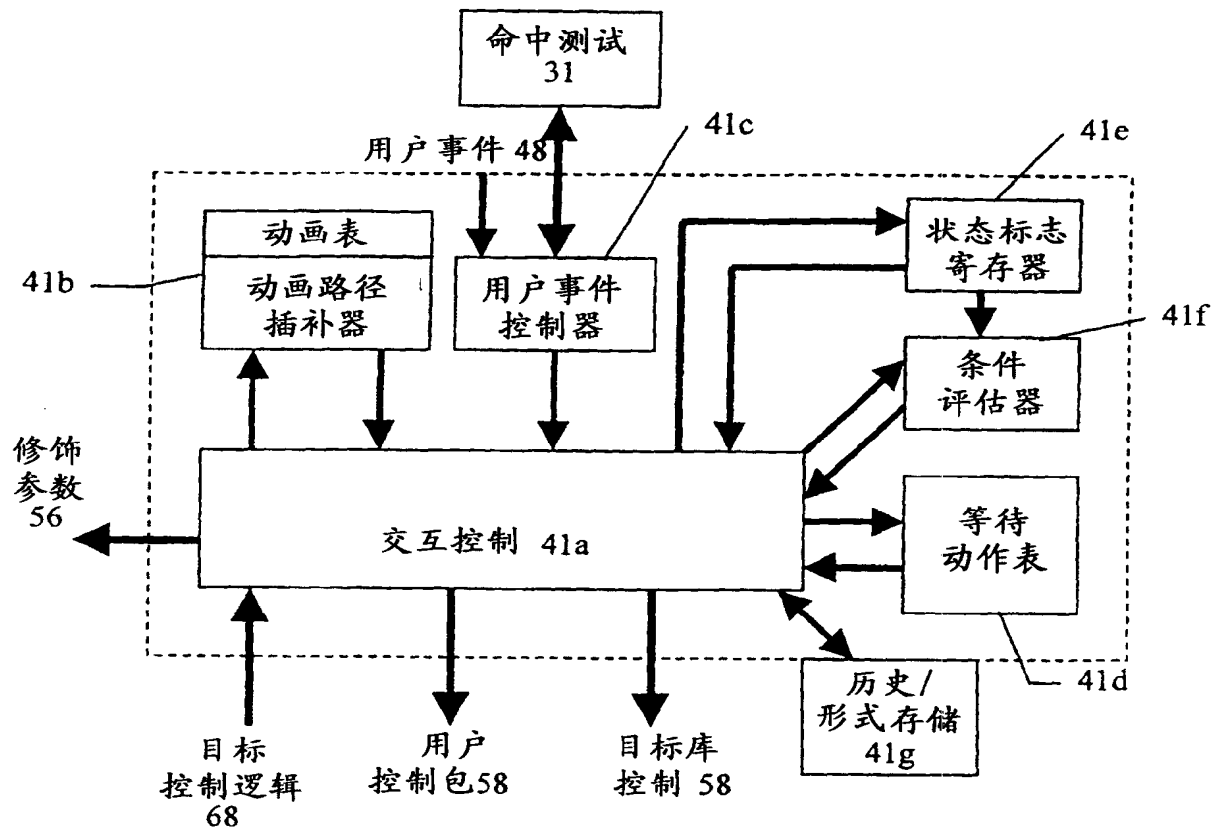


图11

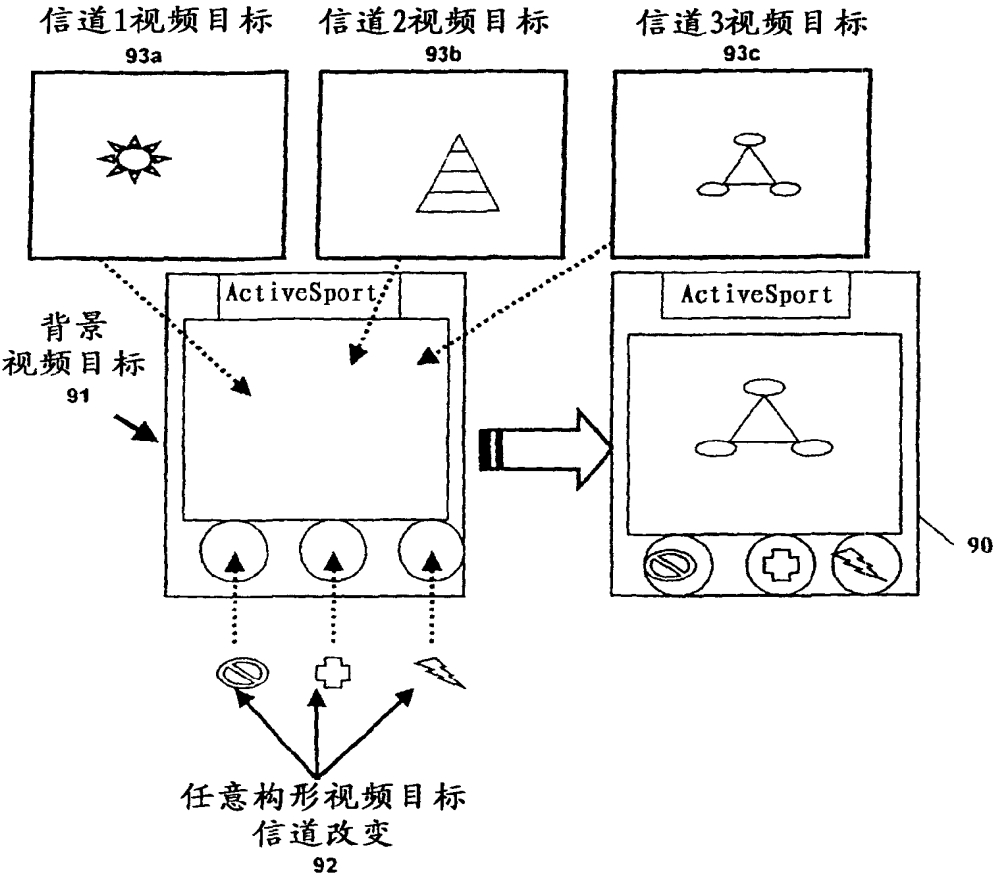


图12

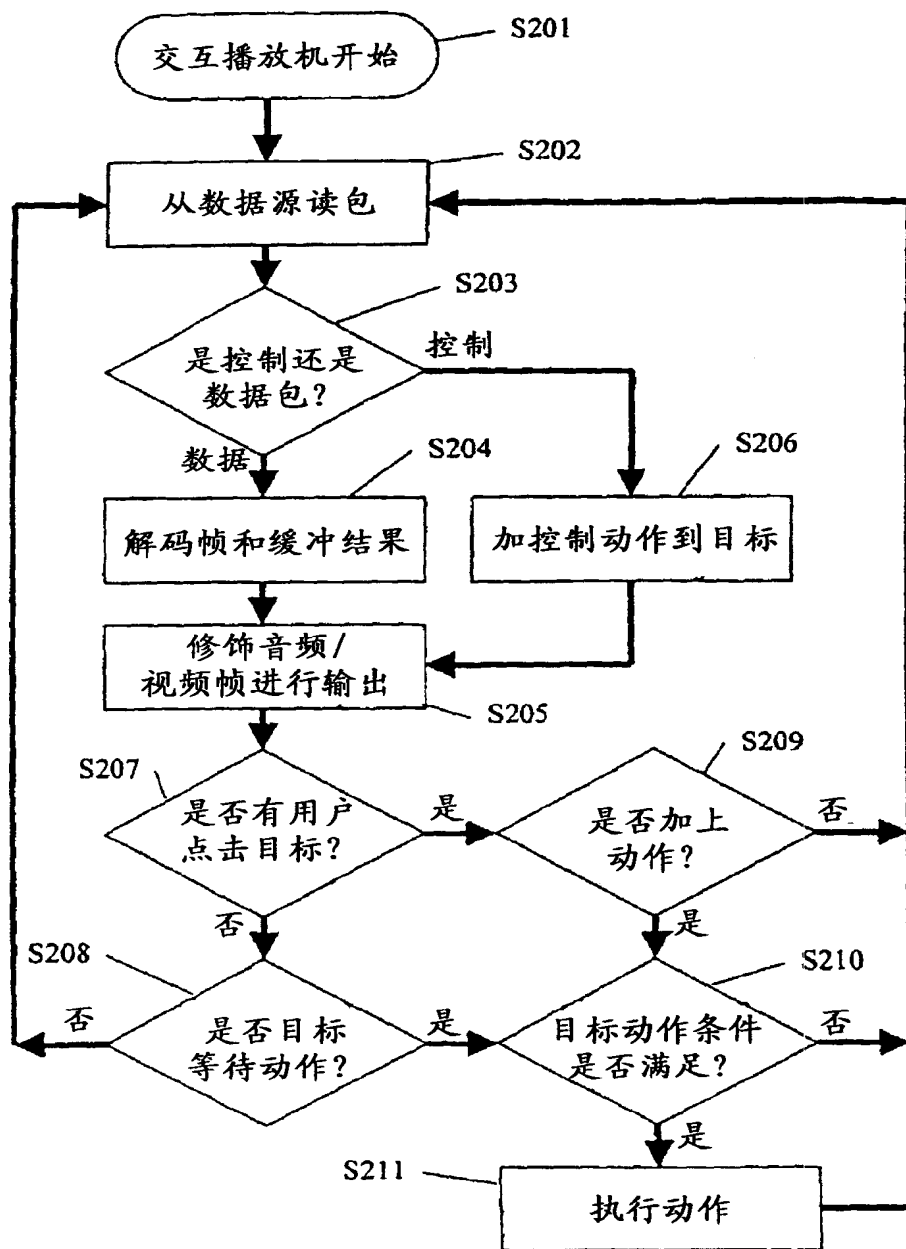


图 13



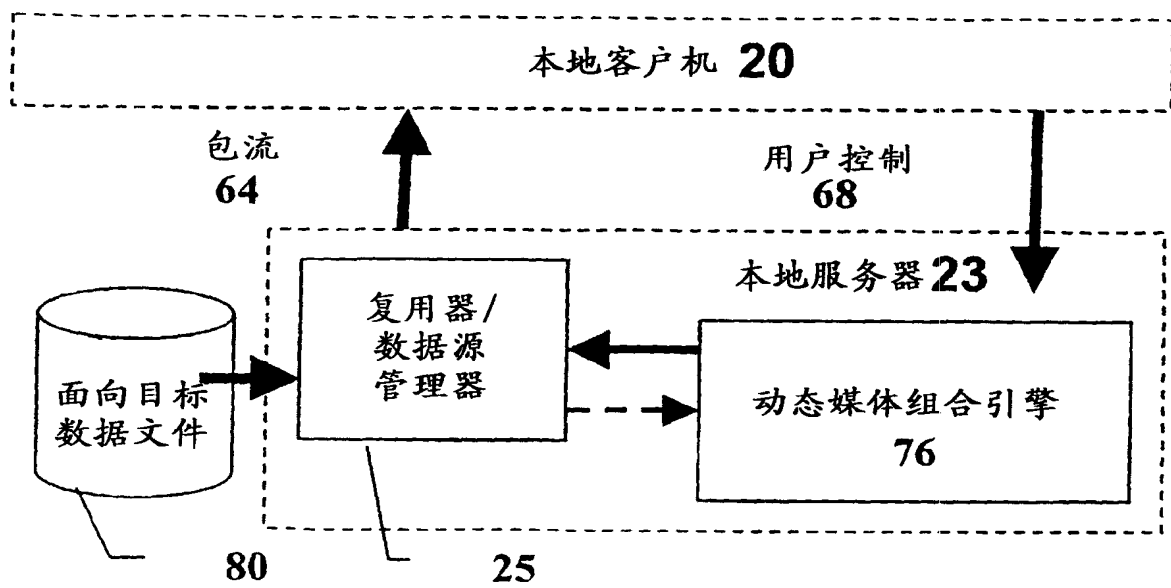


图 14

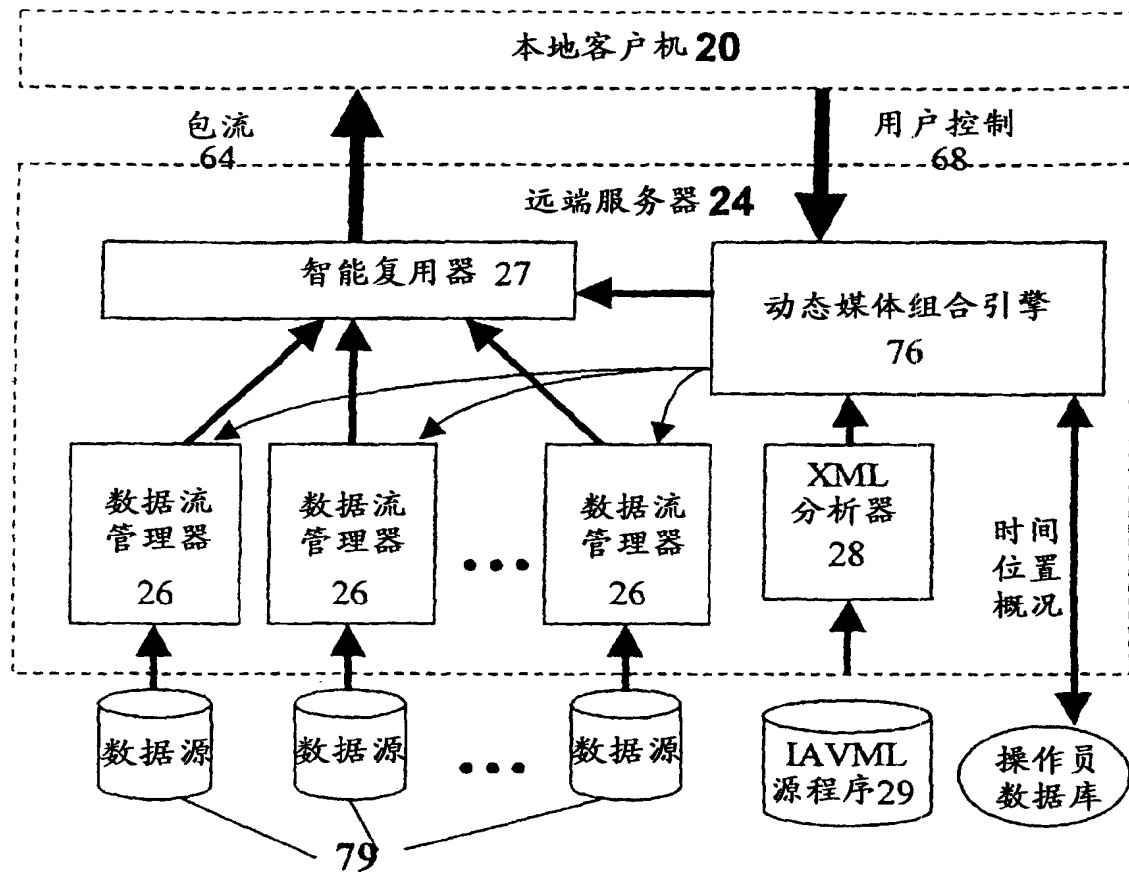


图15

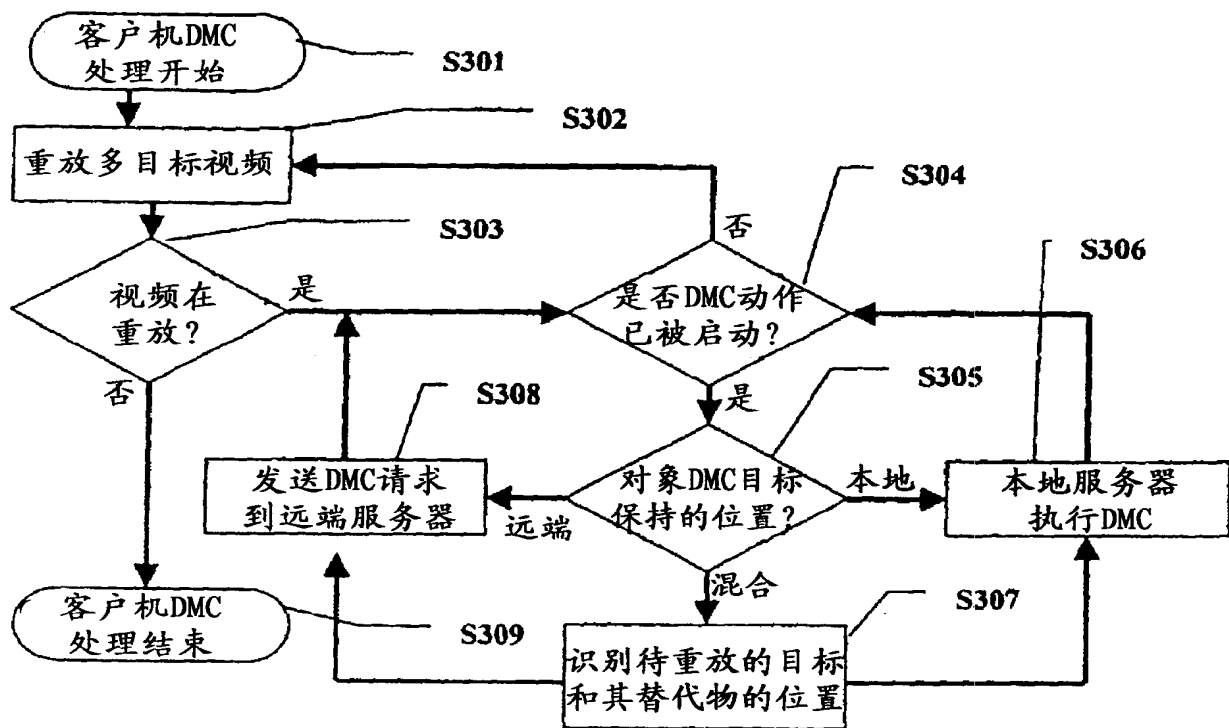


图16

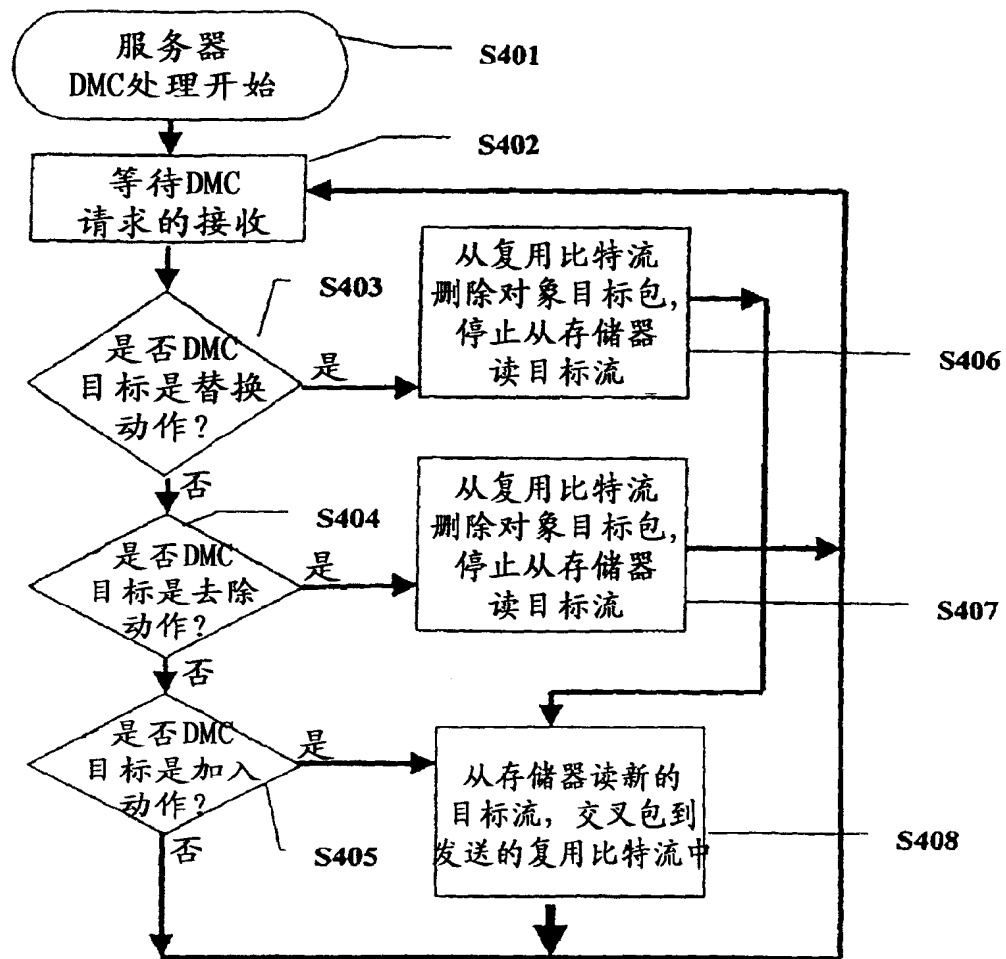


图 17

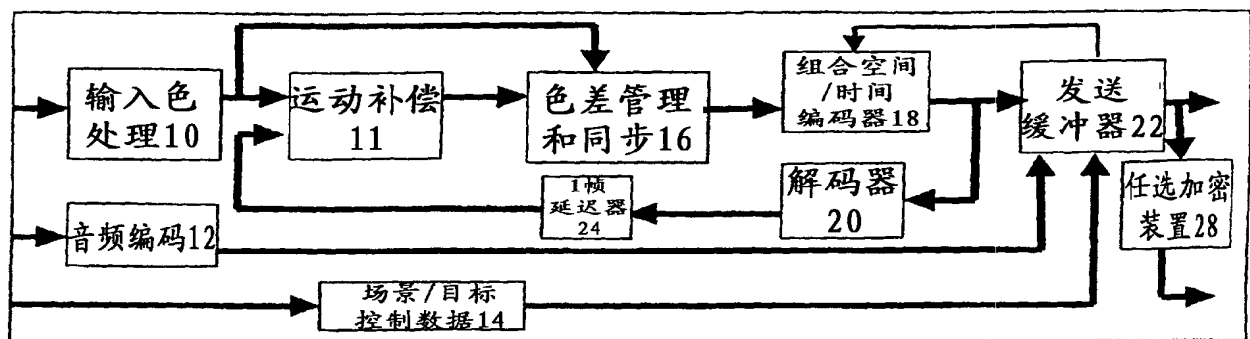


图 18

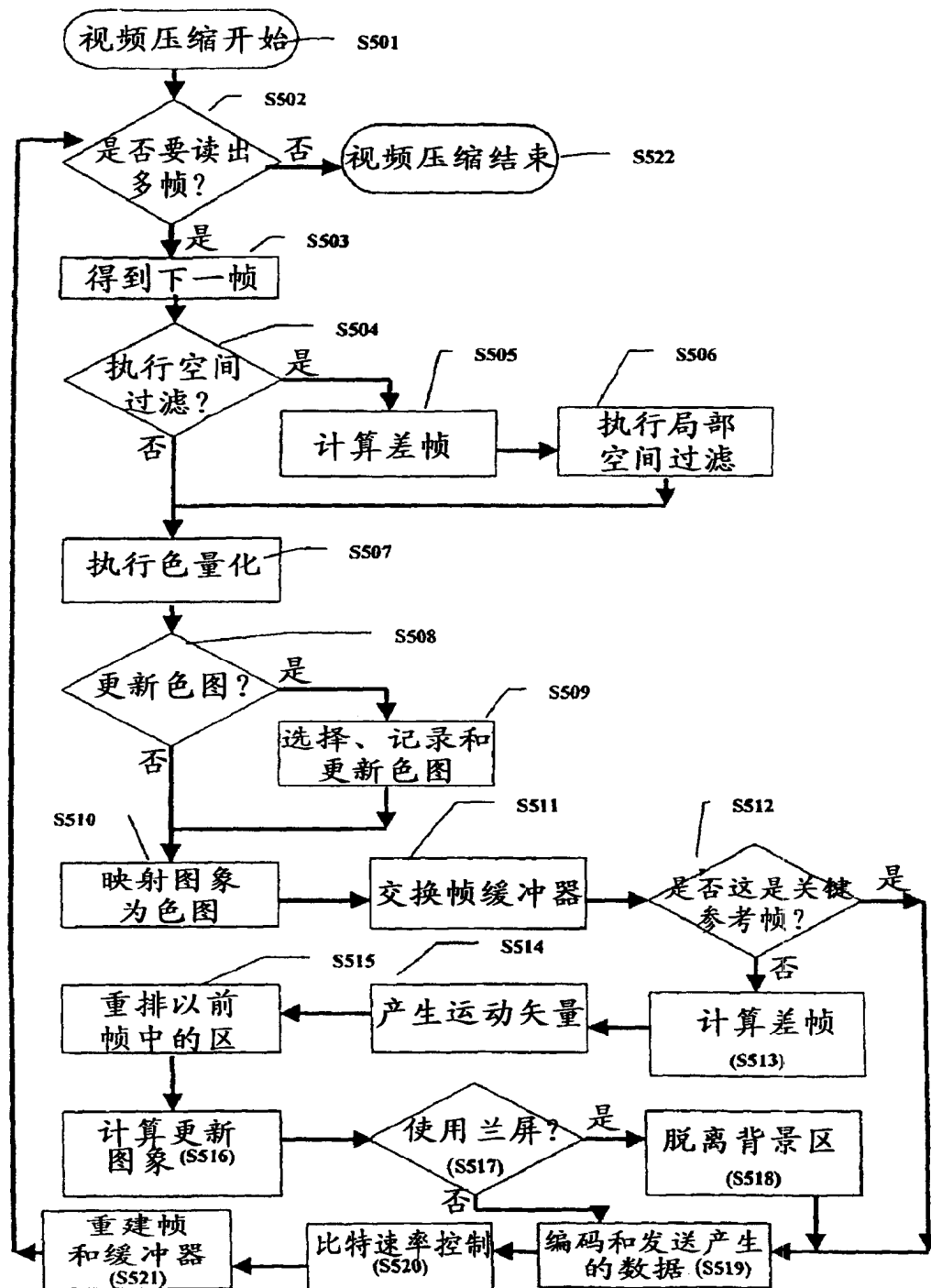


图19

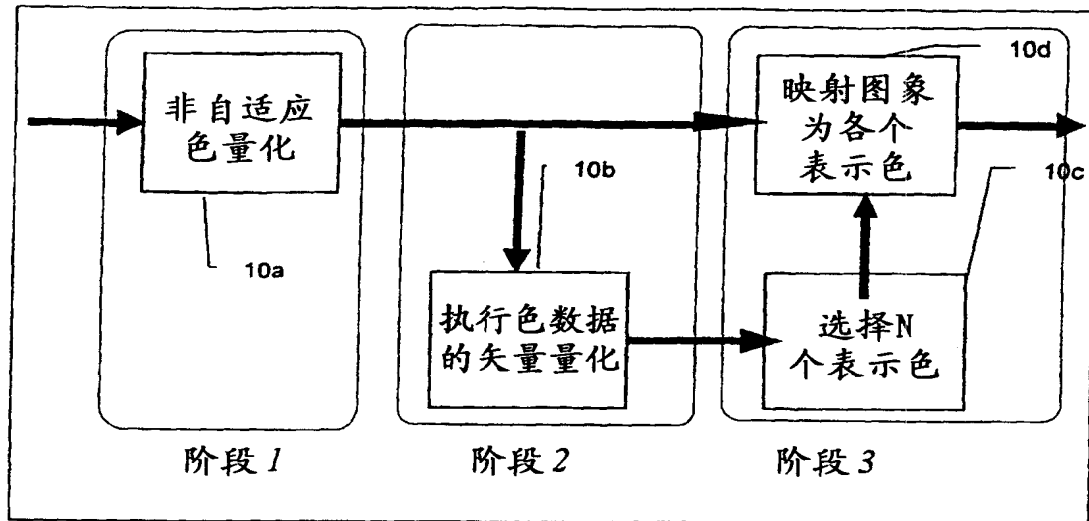


图 20

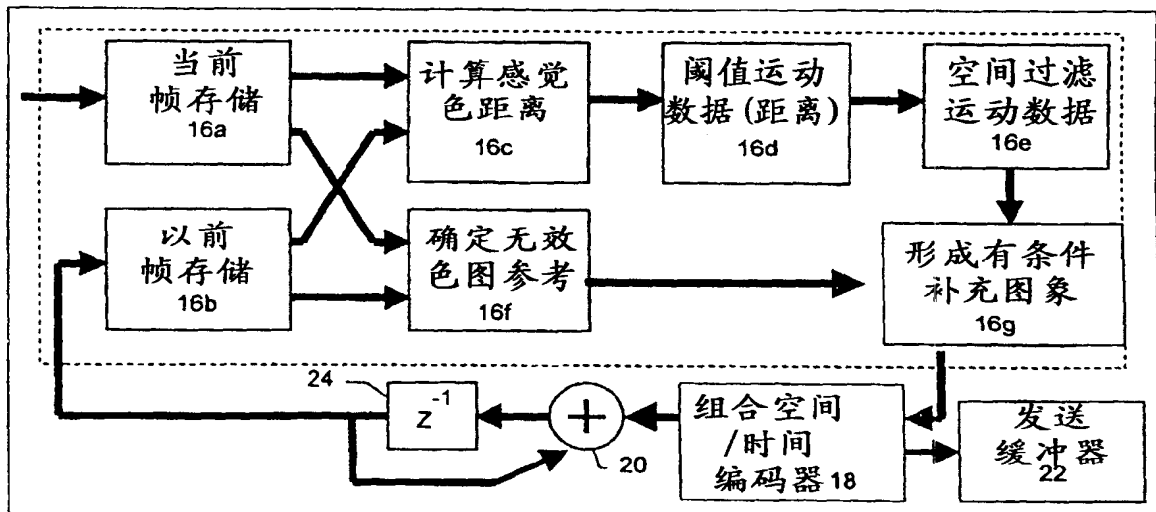


图 21

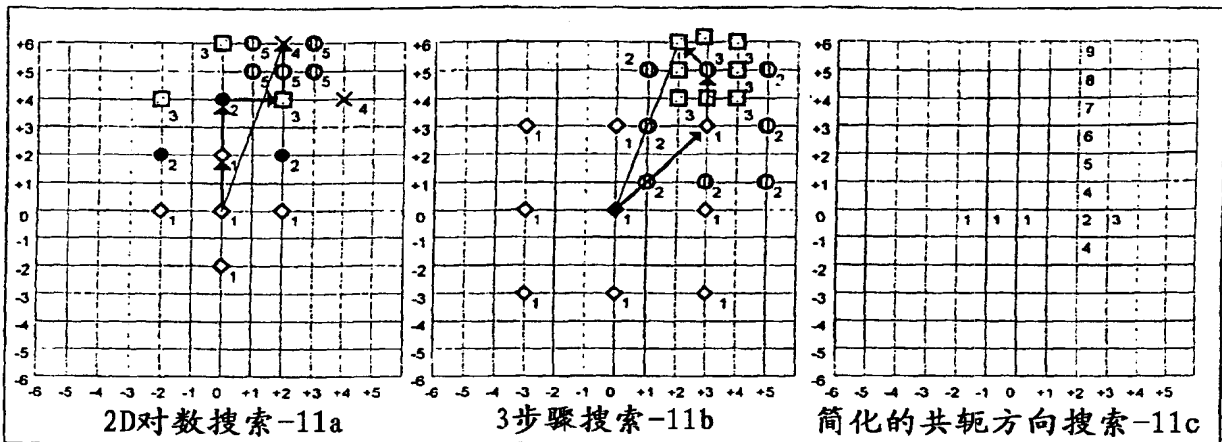


图 22

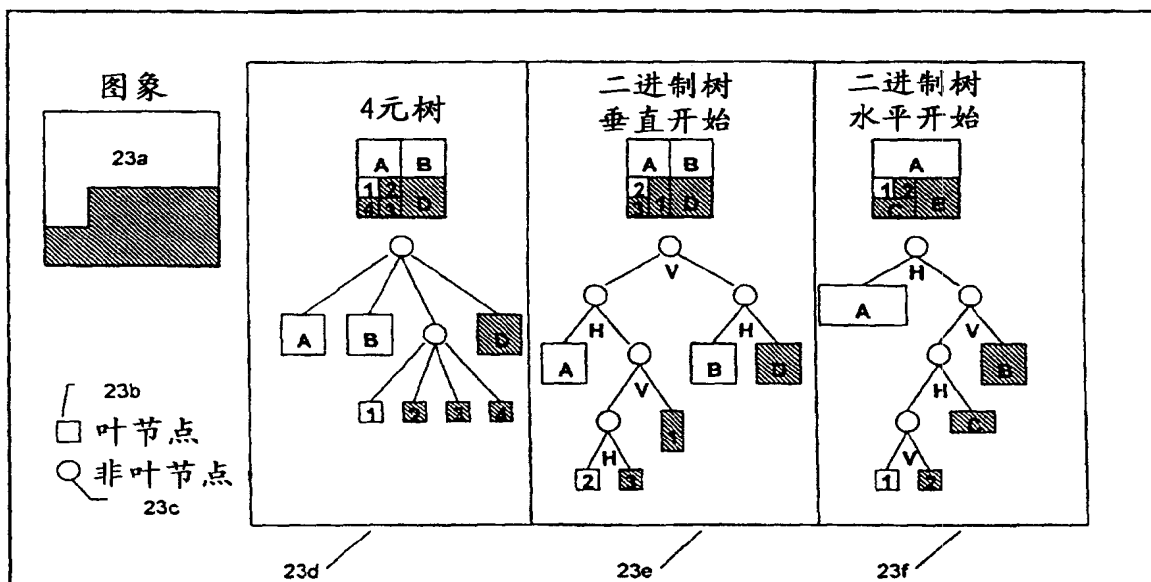


图 23



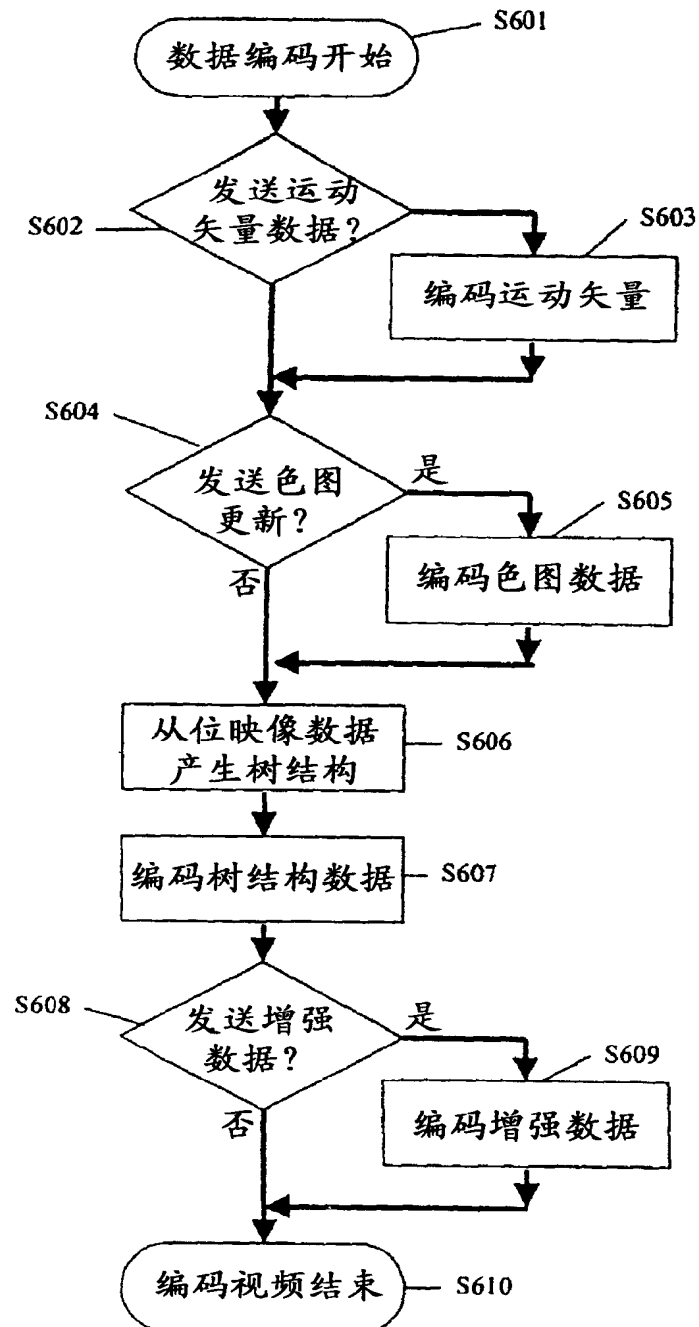
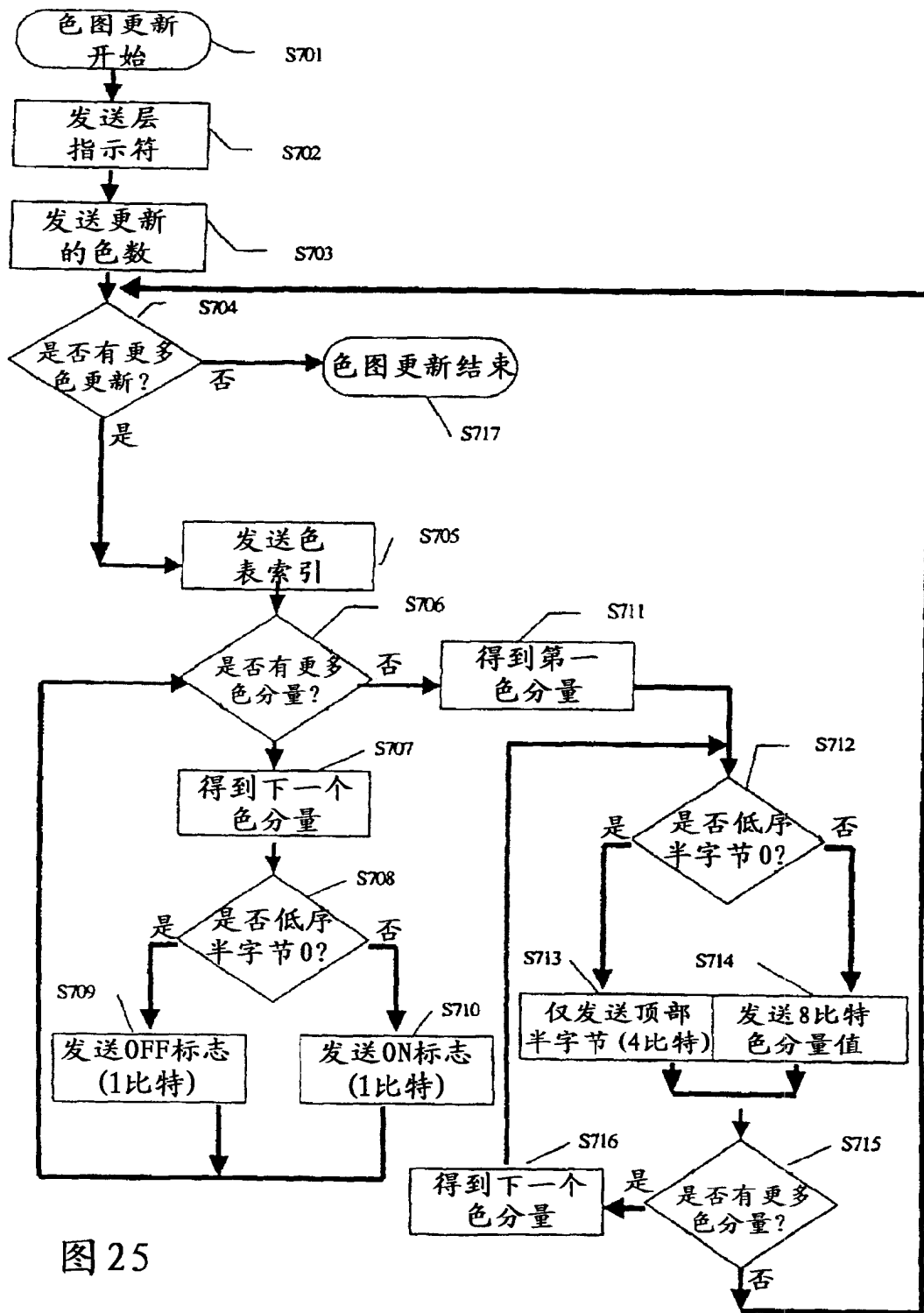


图 24



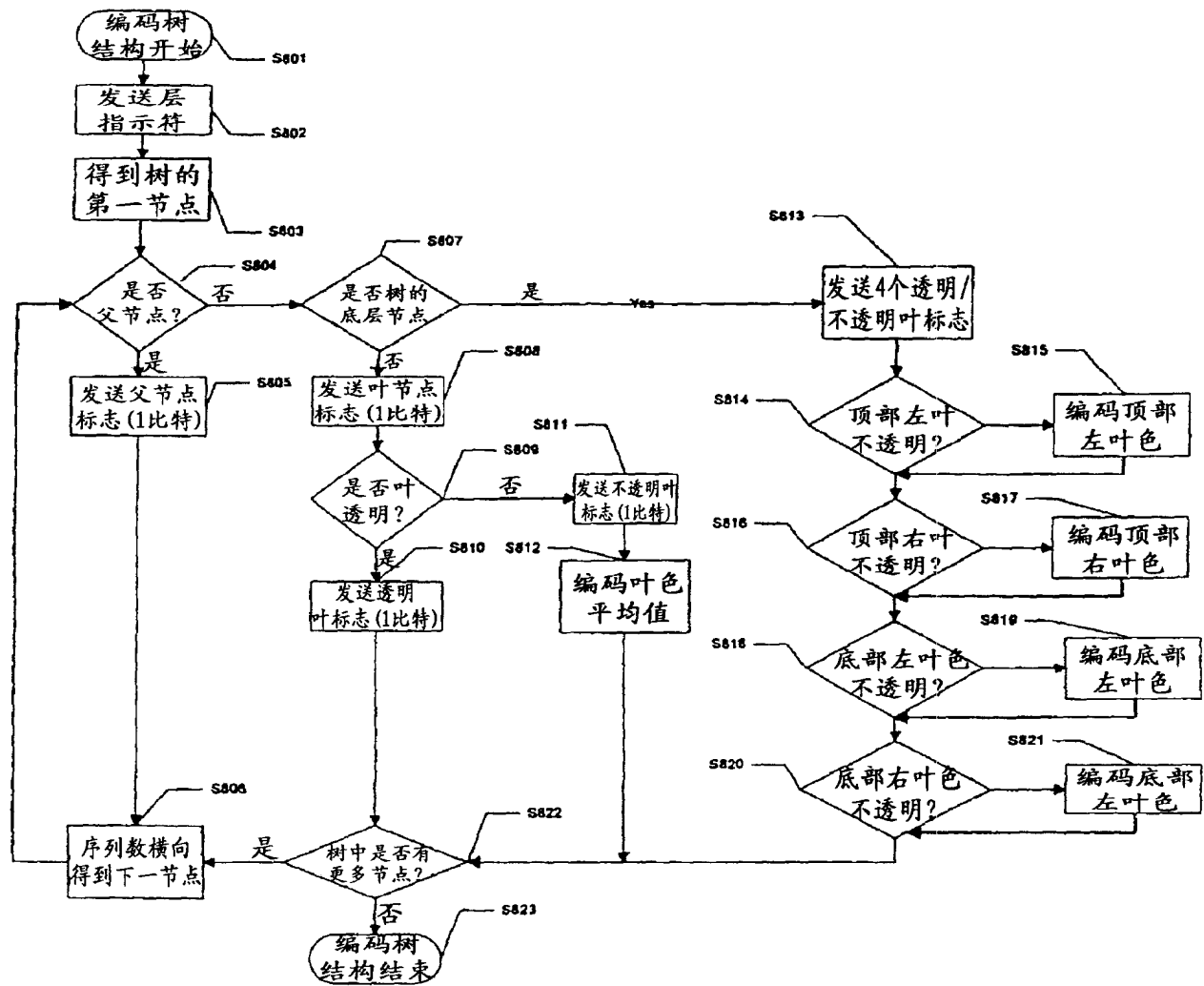


图 26

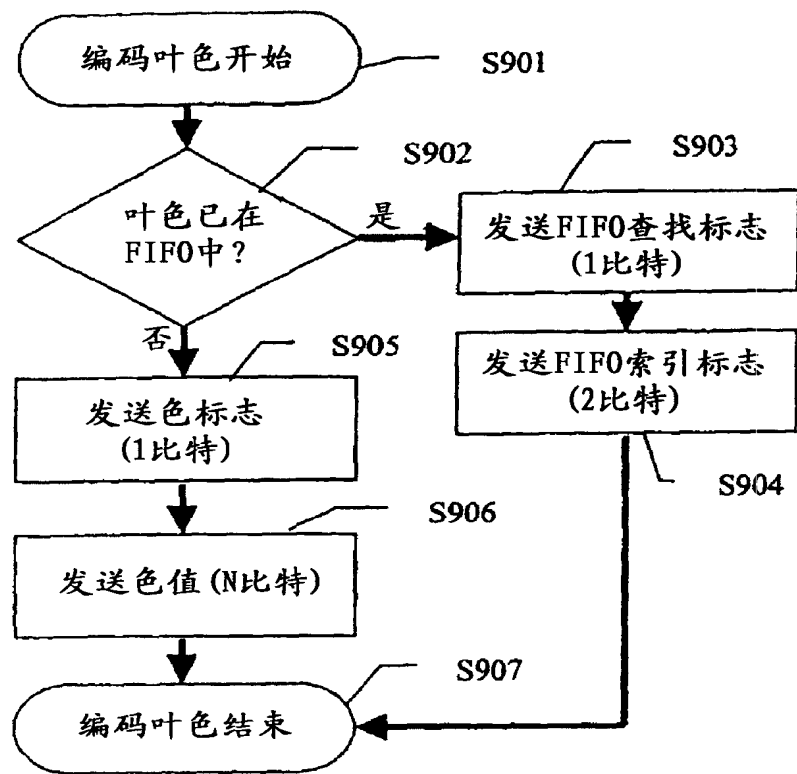


图 27

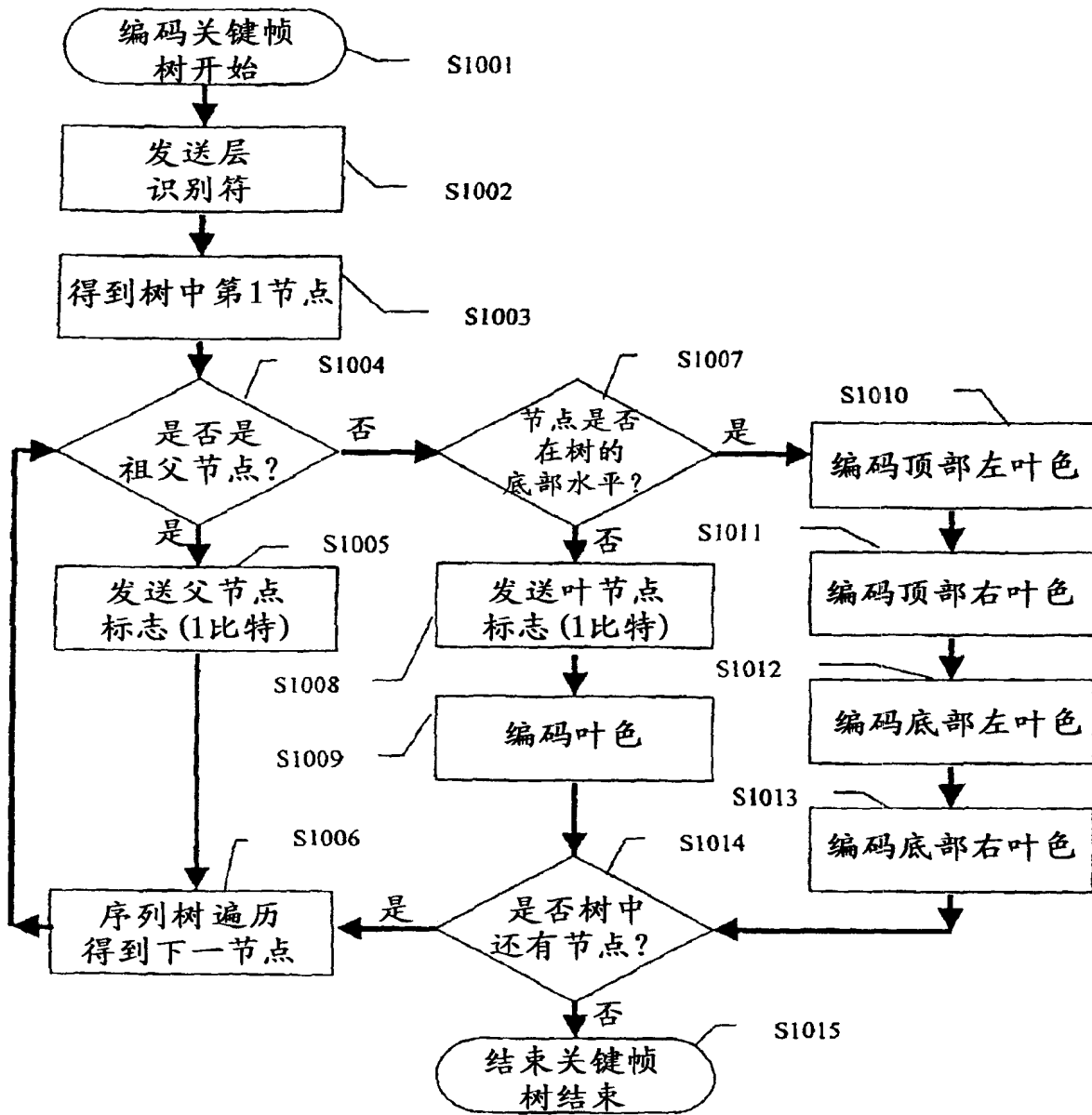


图28

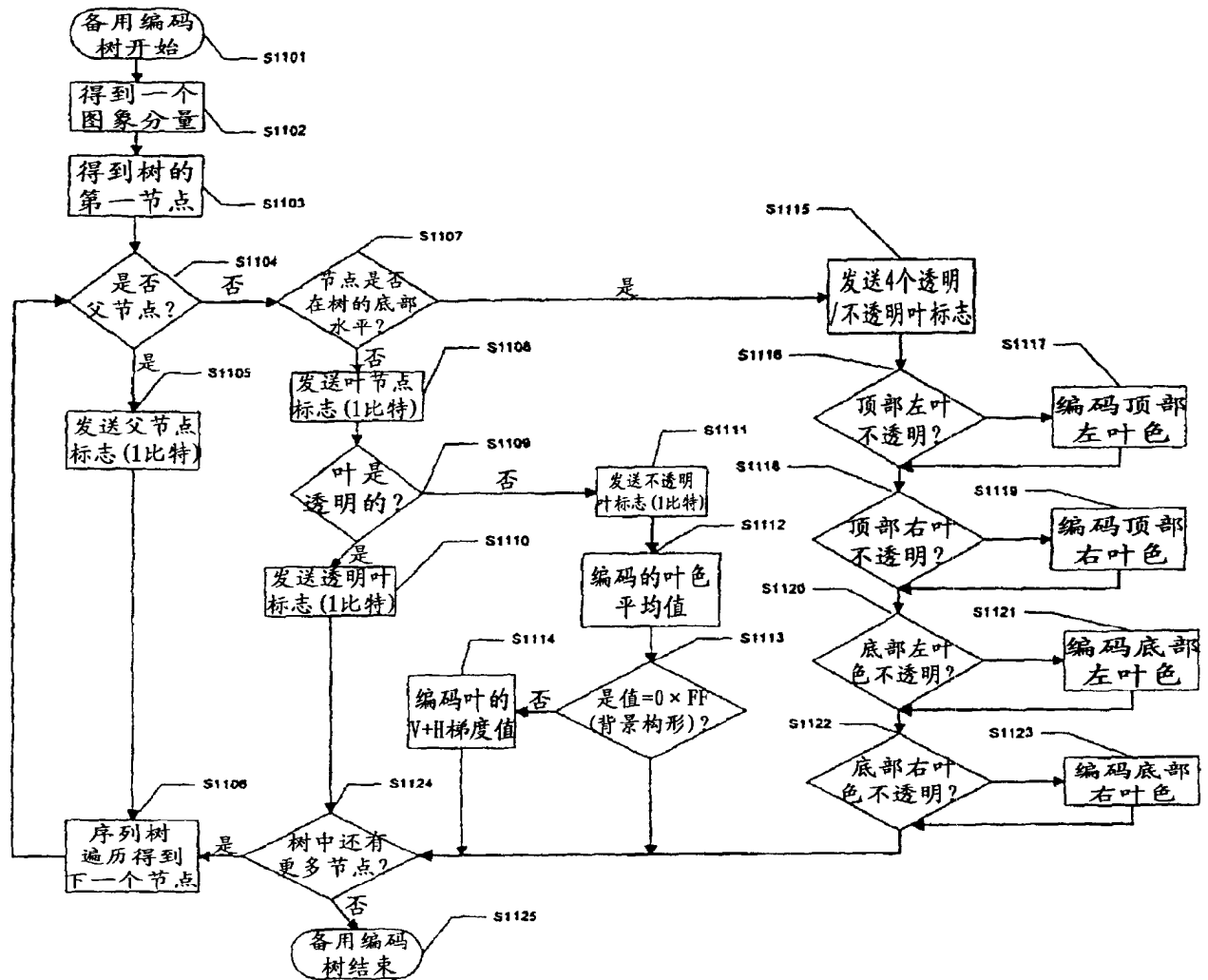


图 29

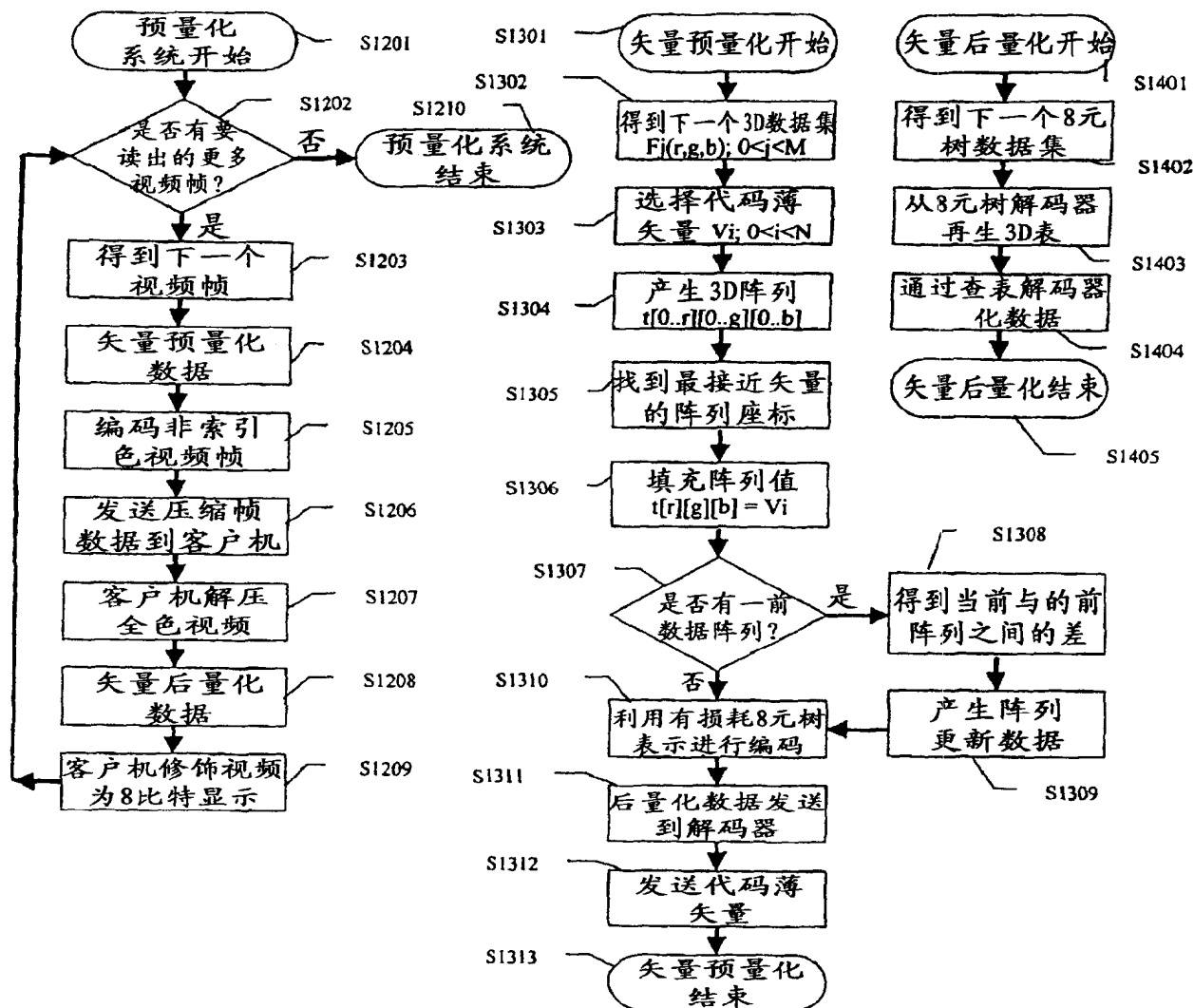


图 30

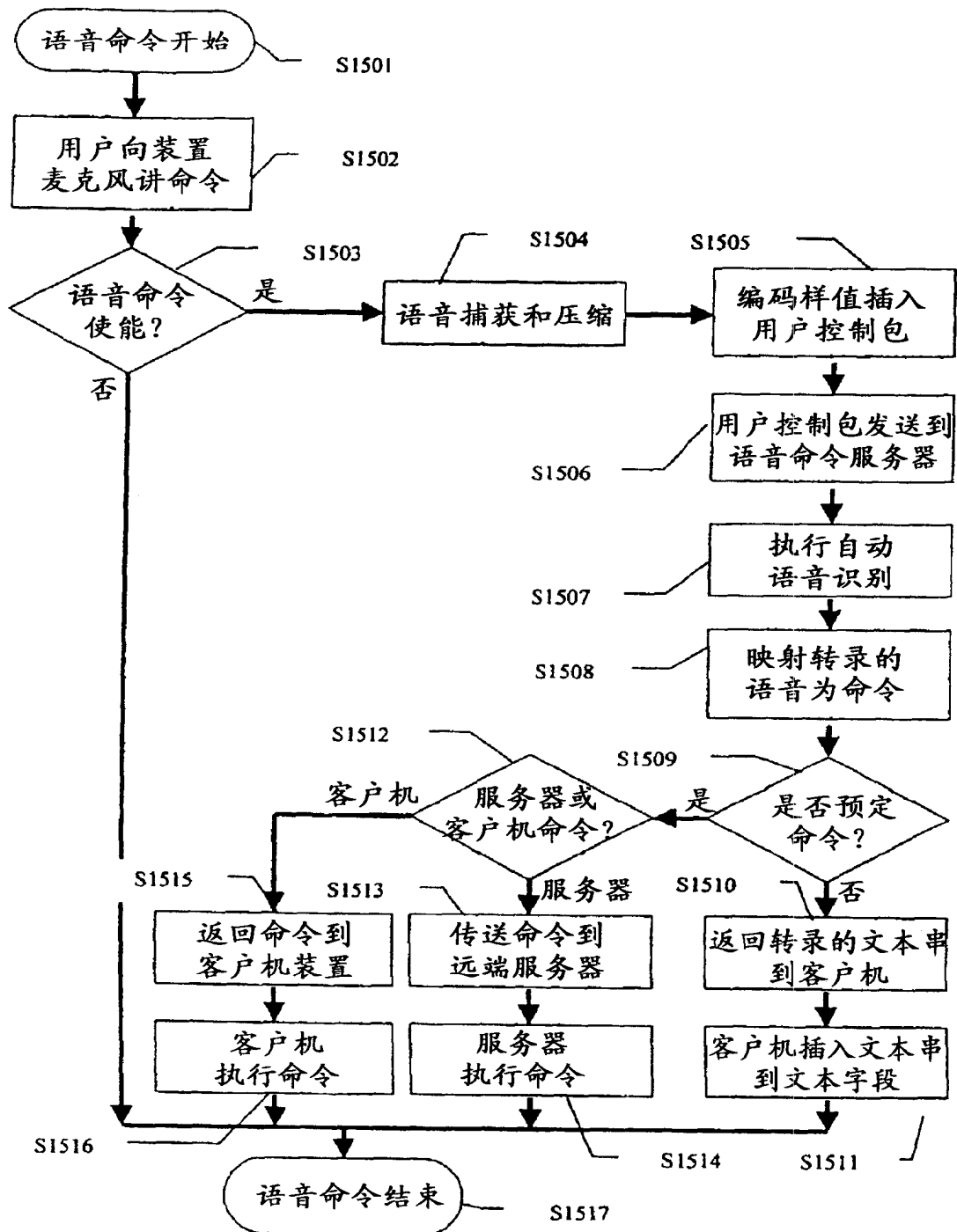


图 31



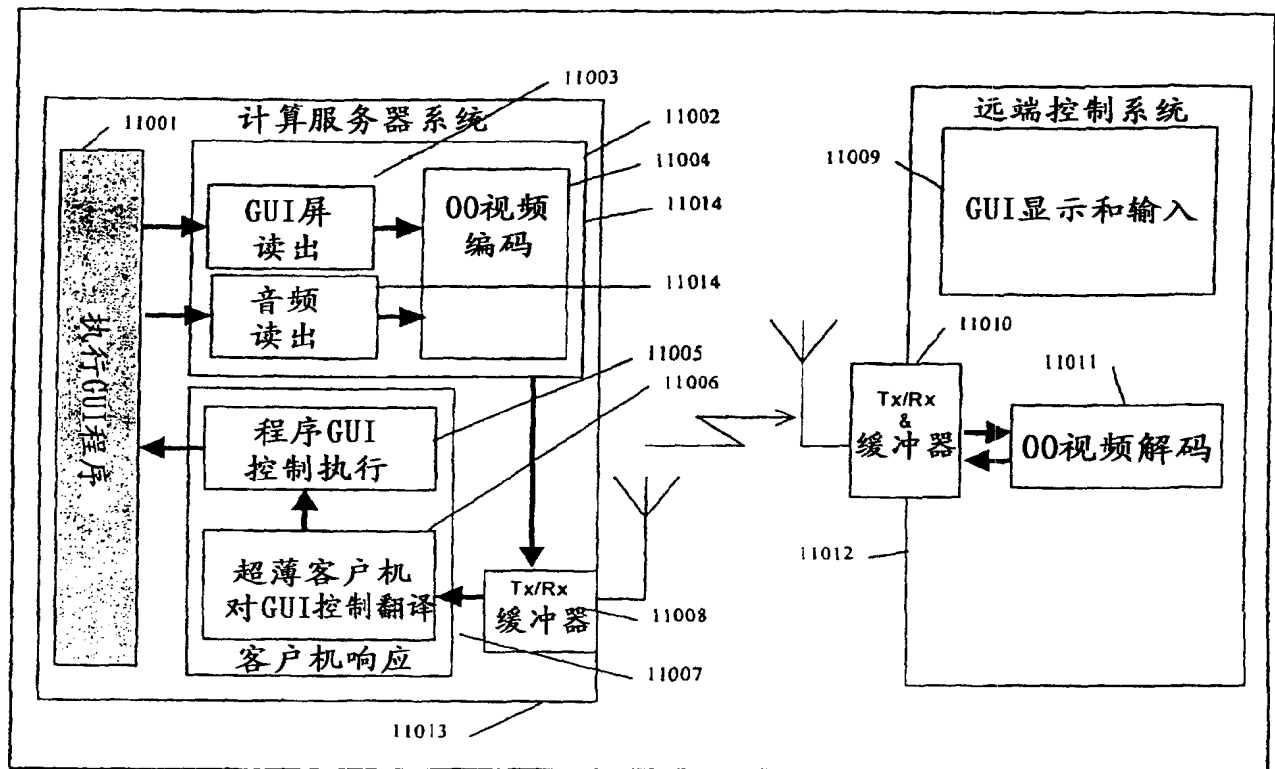
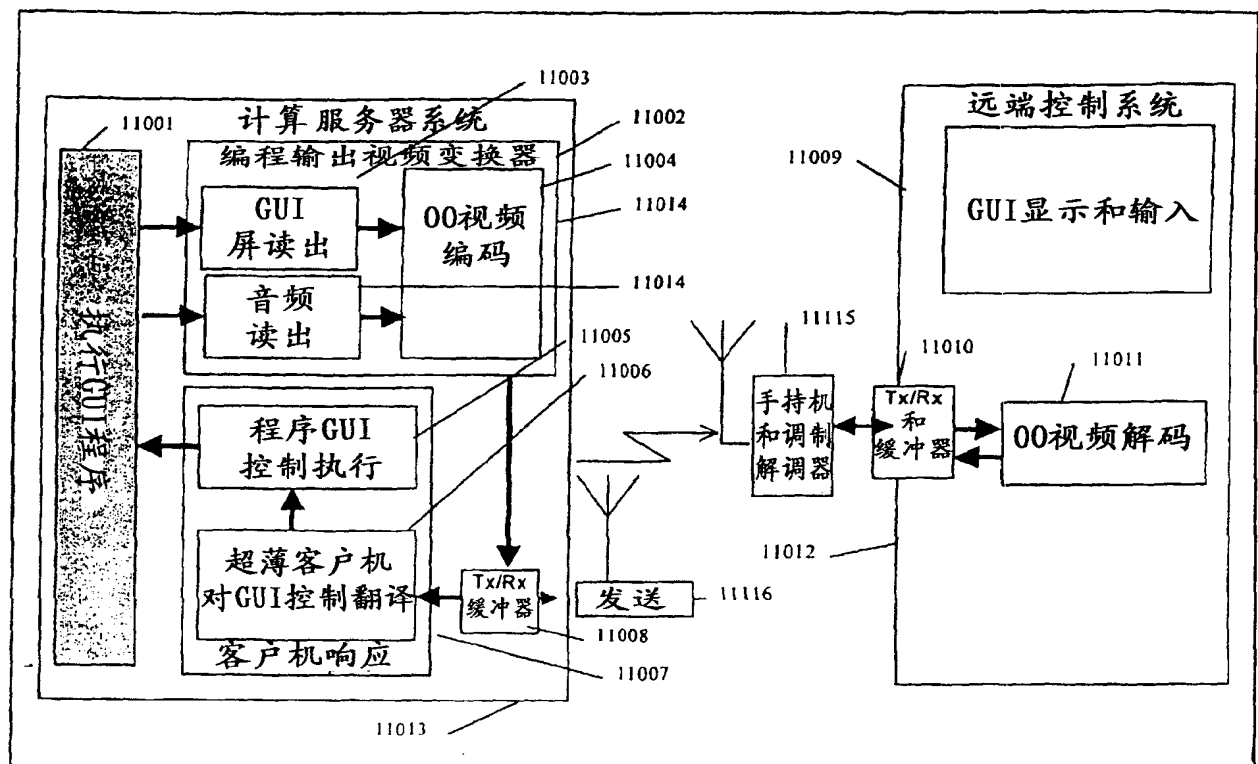


图 32



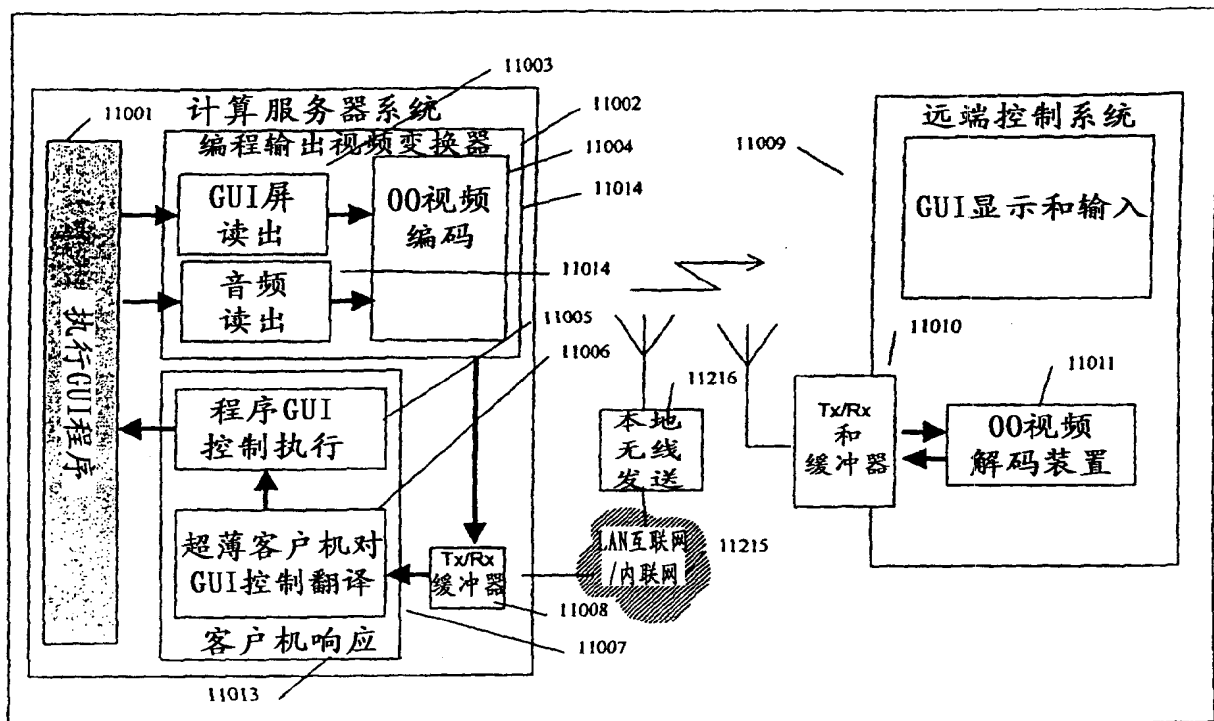


图 34

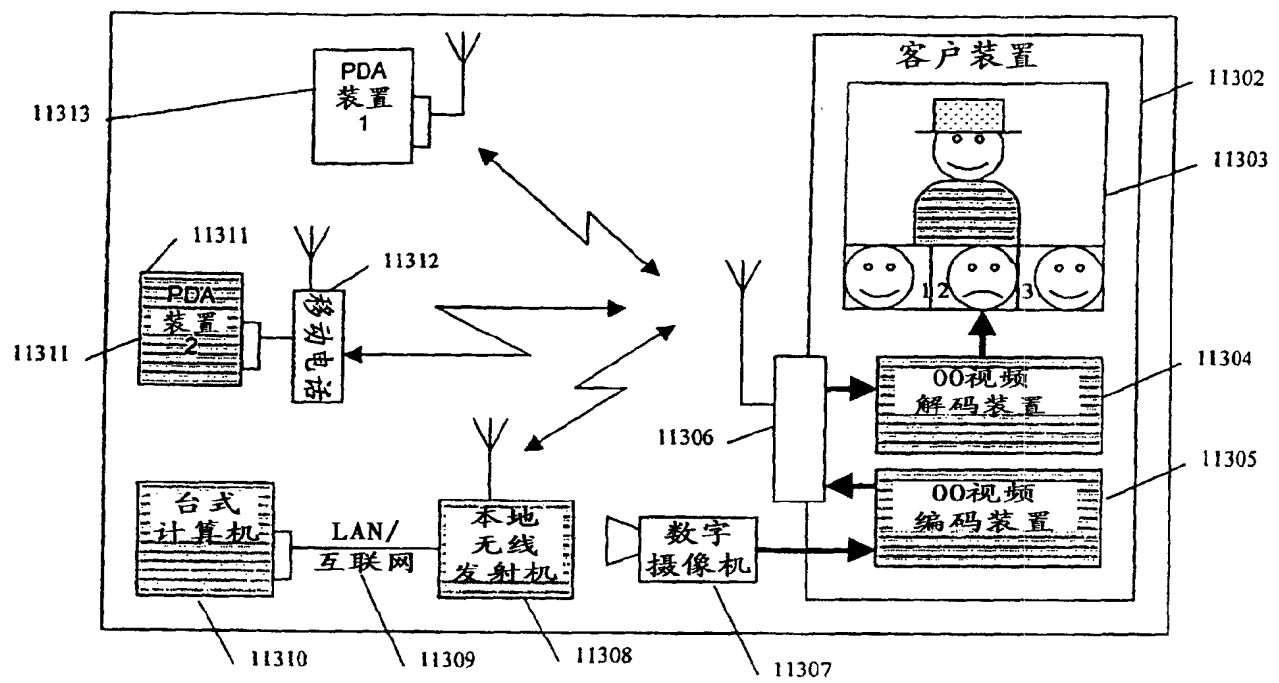


图 35

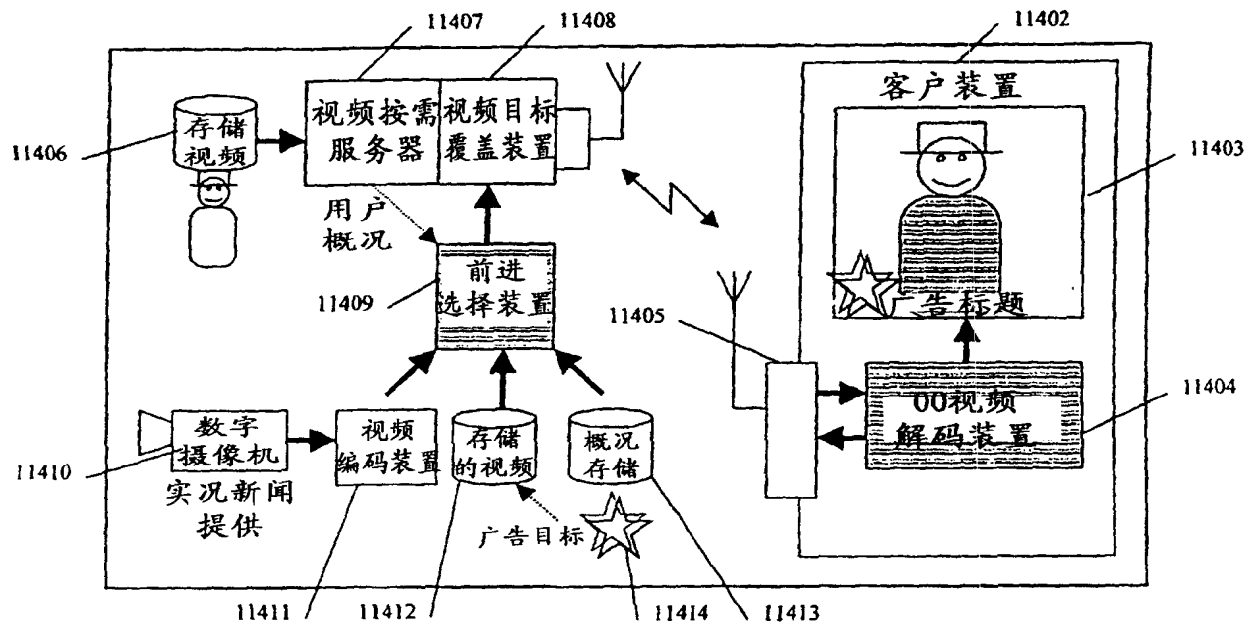


图 36

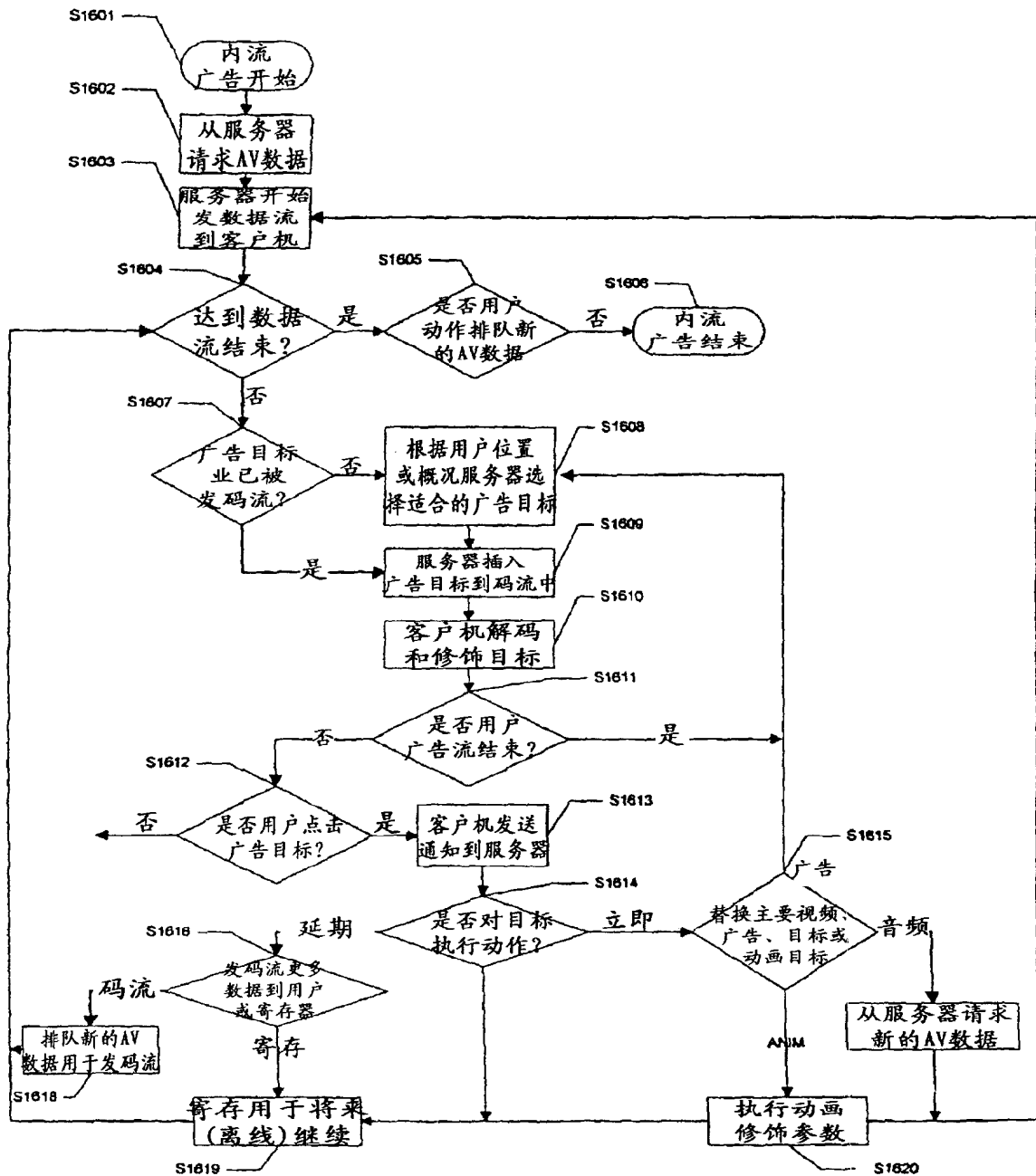
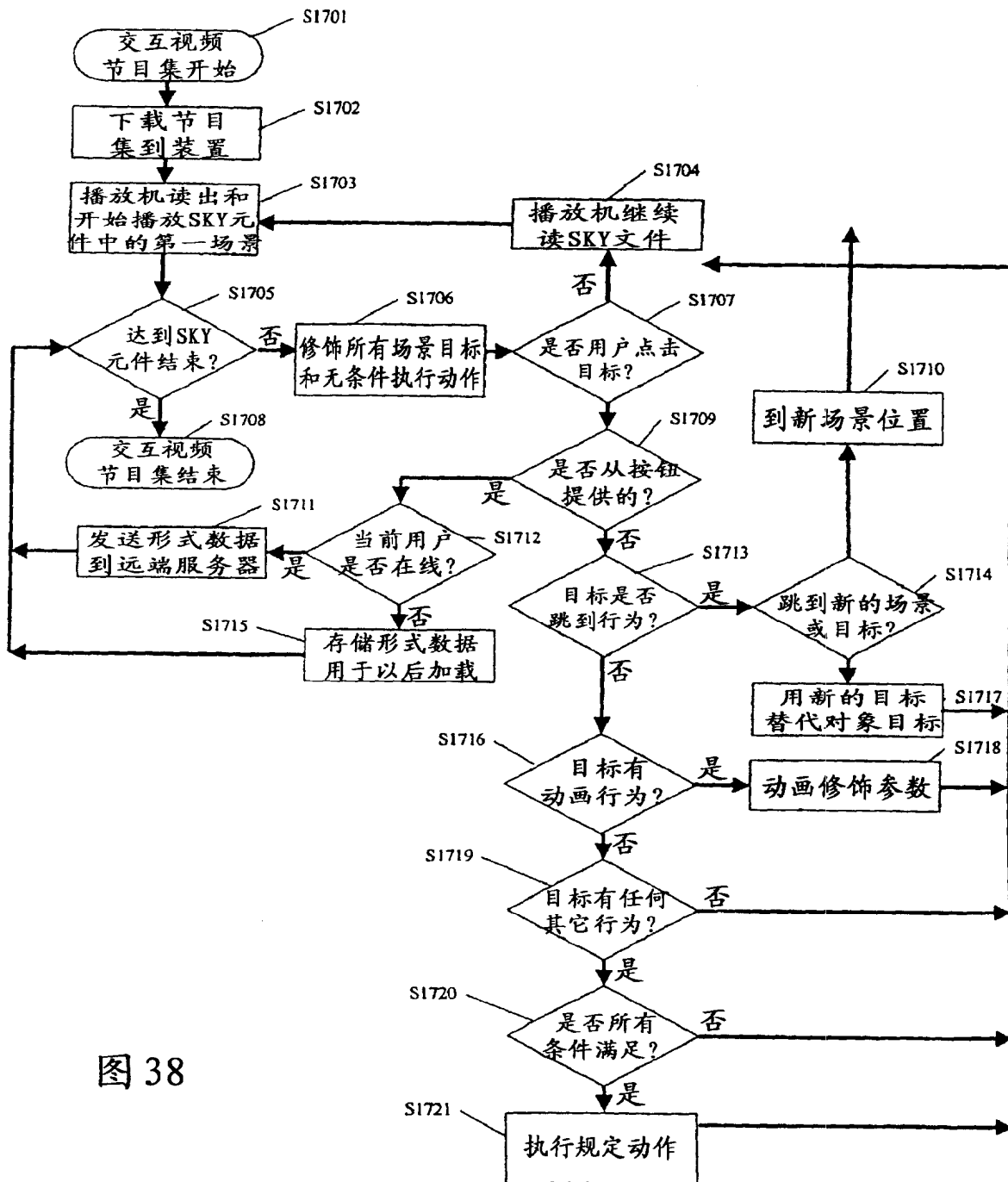


图 37



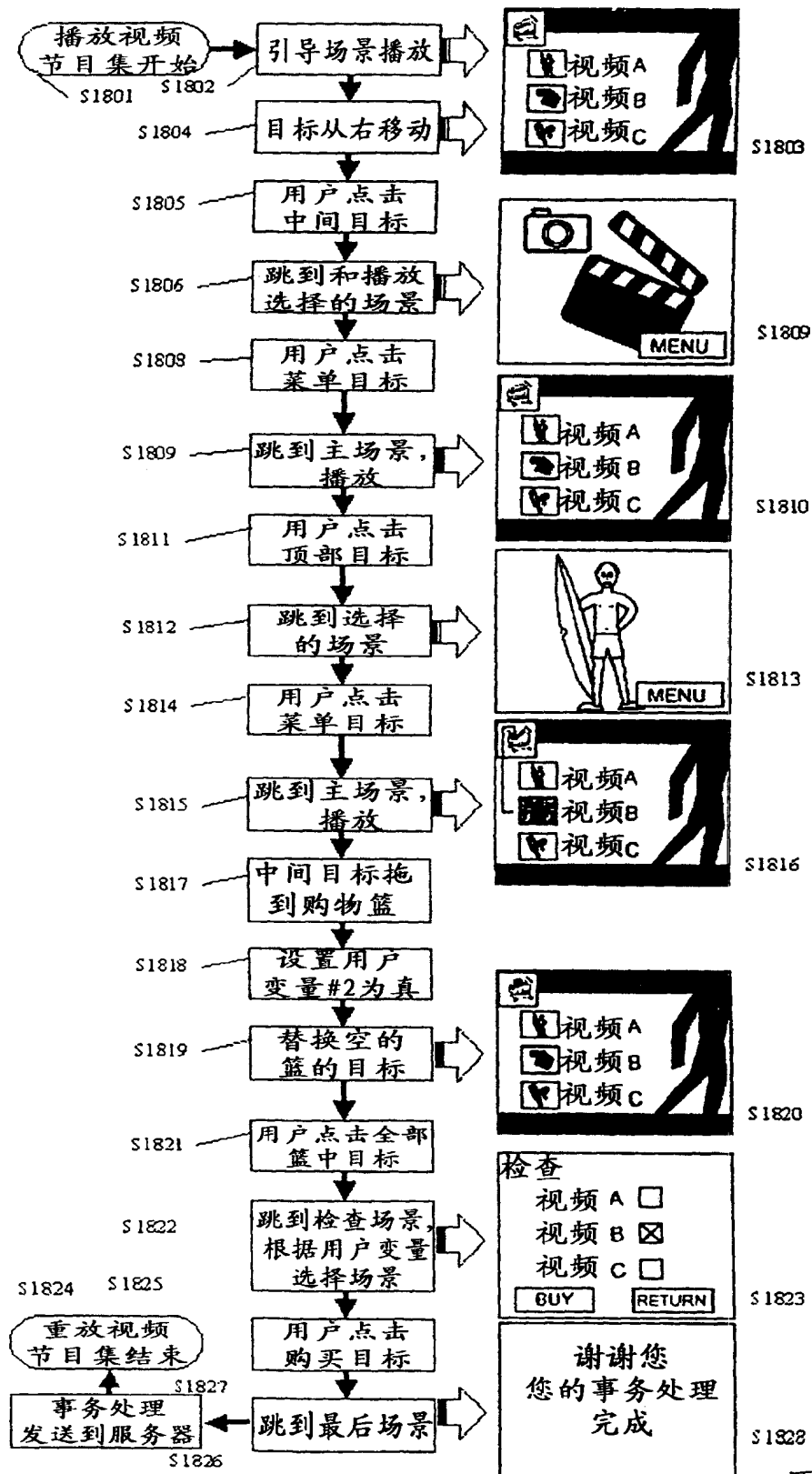


图 39



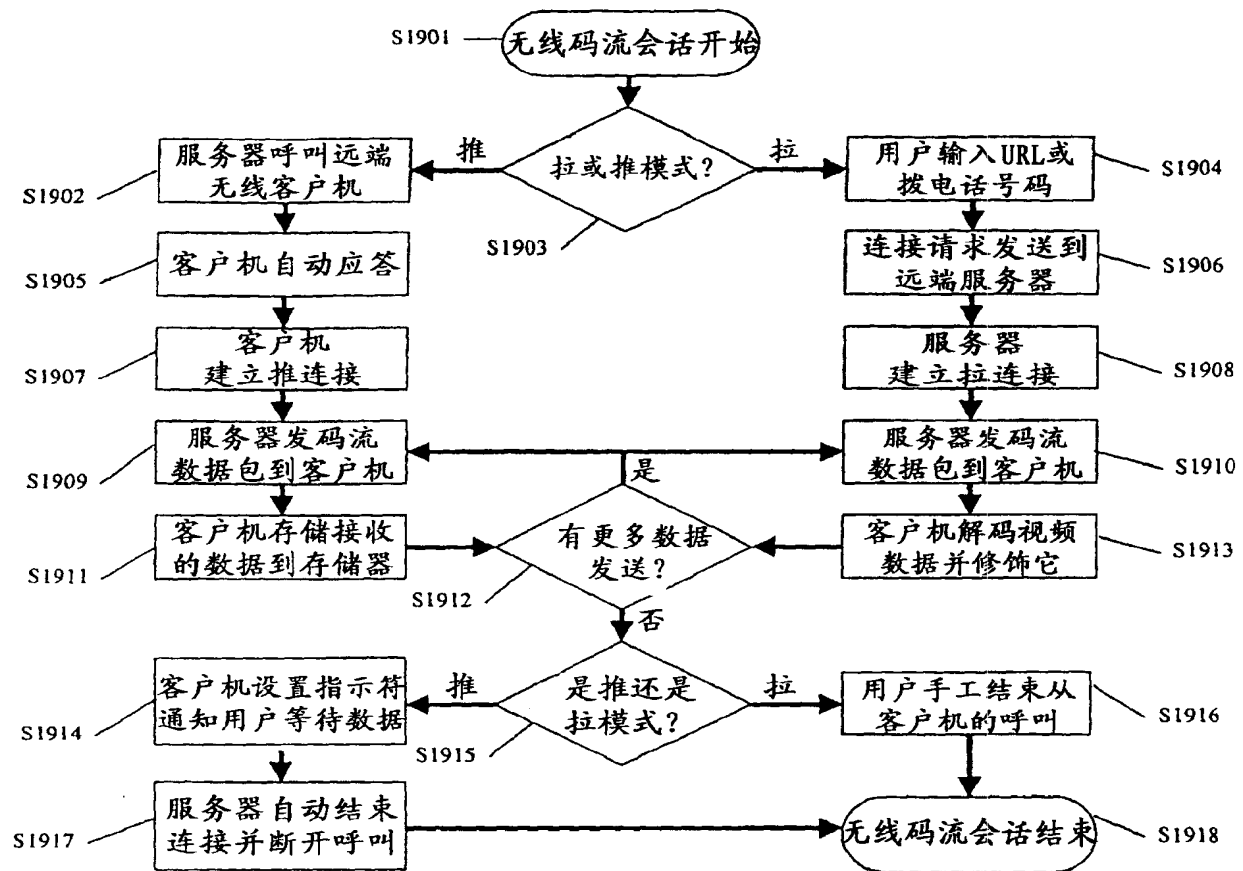


图 40

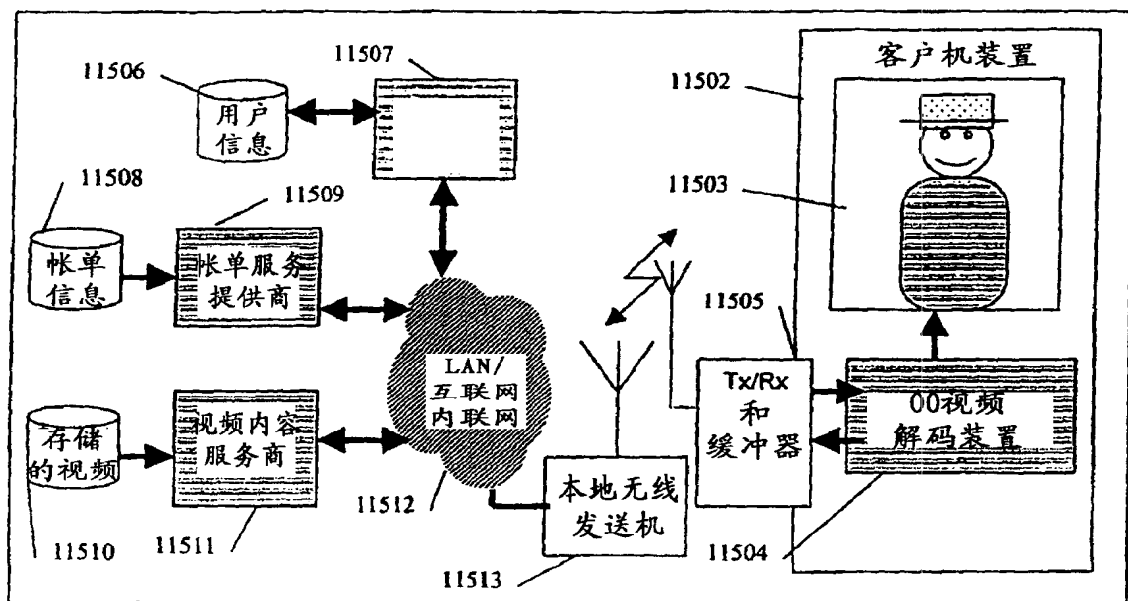


图 41

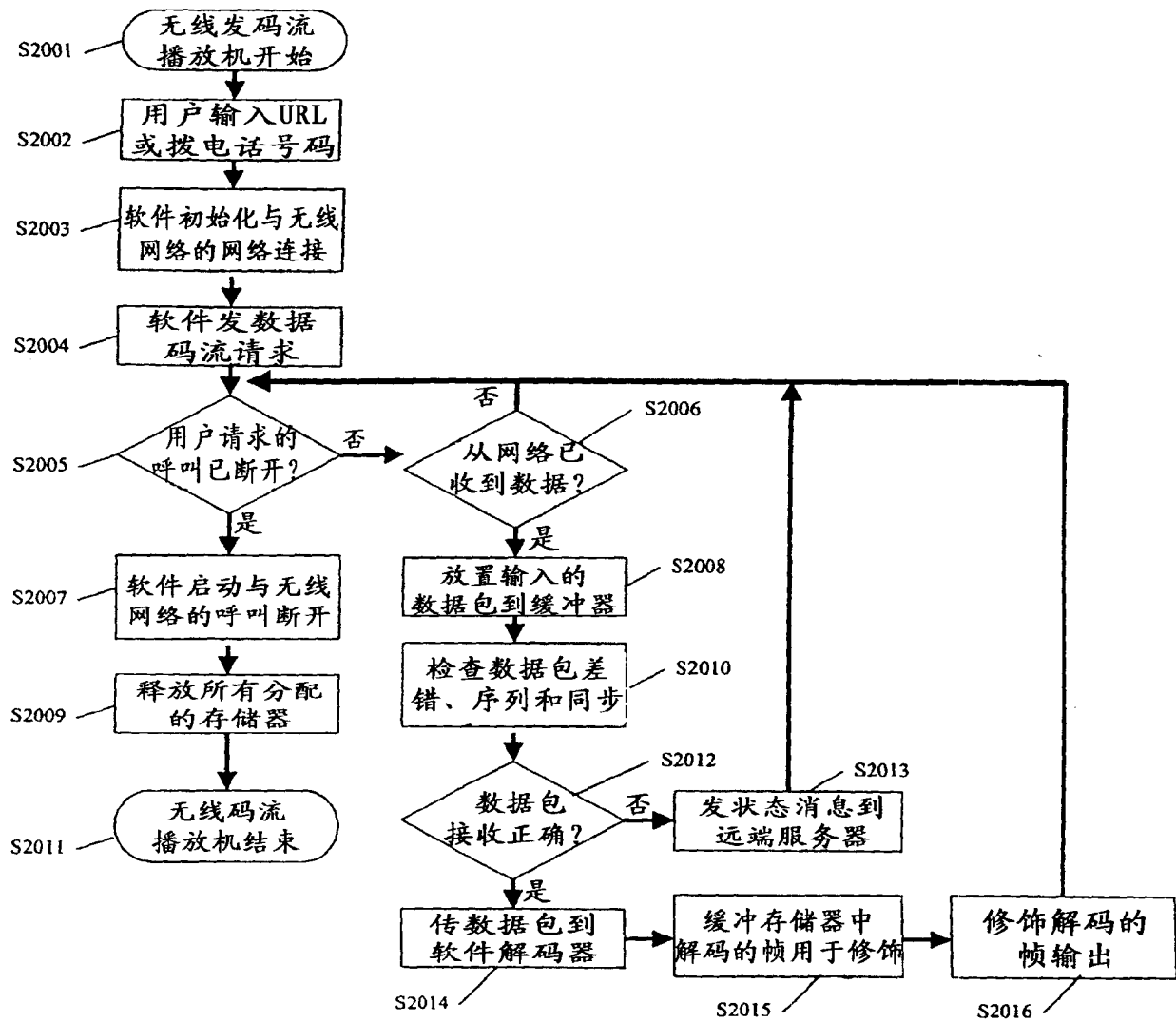


图 42

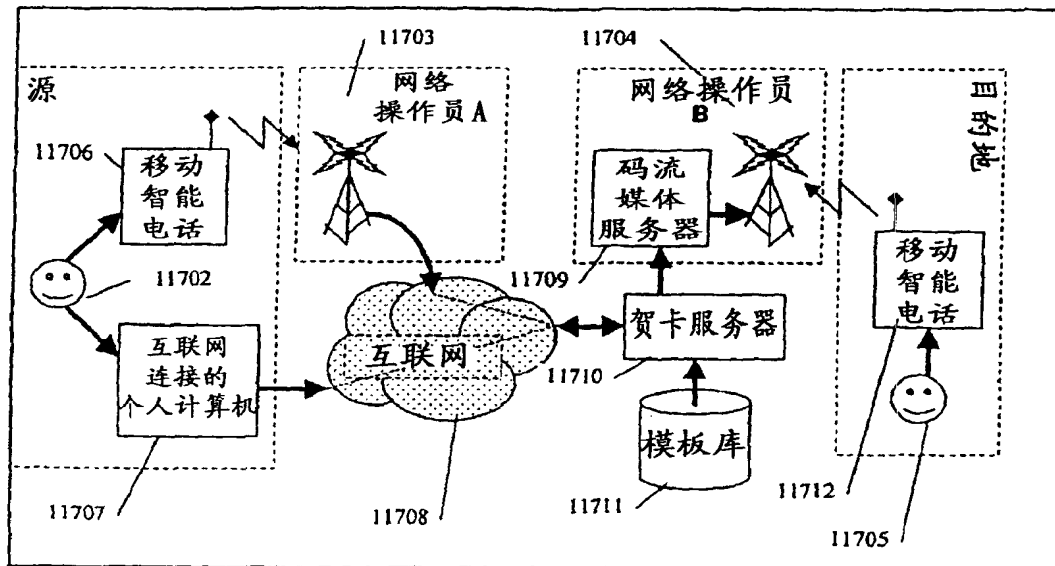


图 44

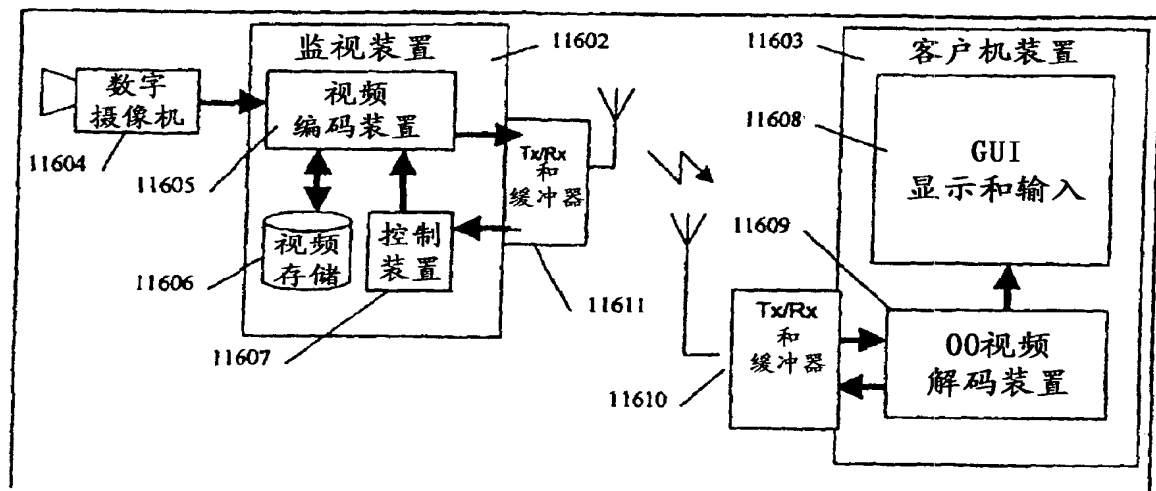


图 43

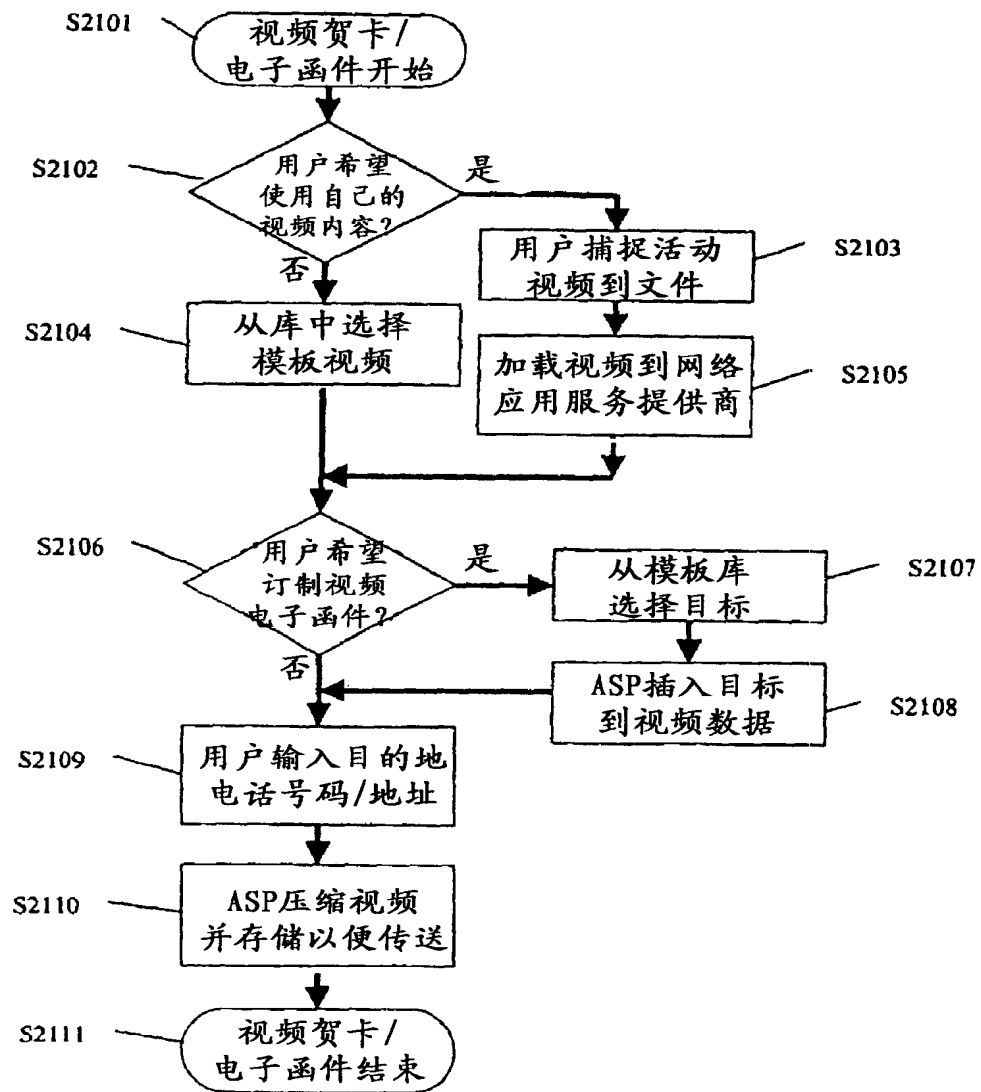


图 45

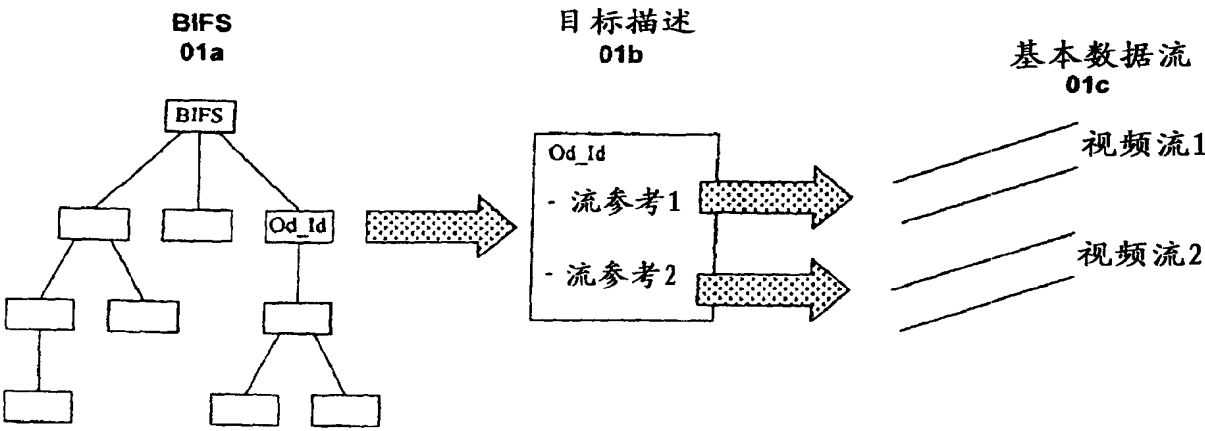


图 46

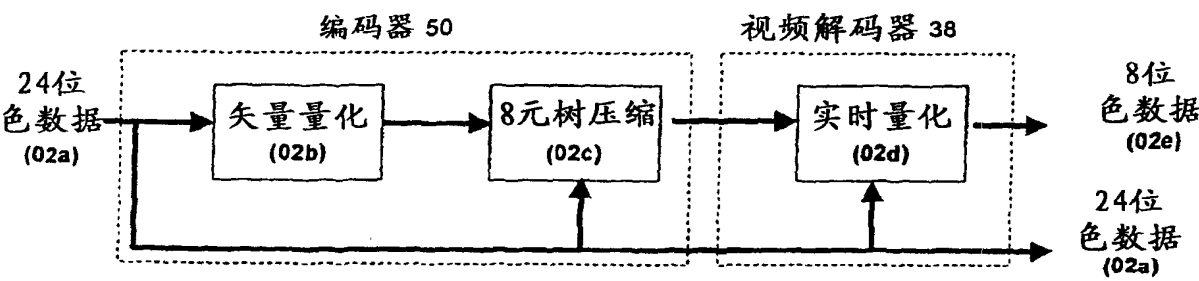


图 47



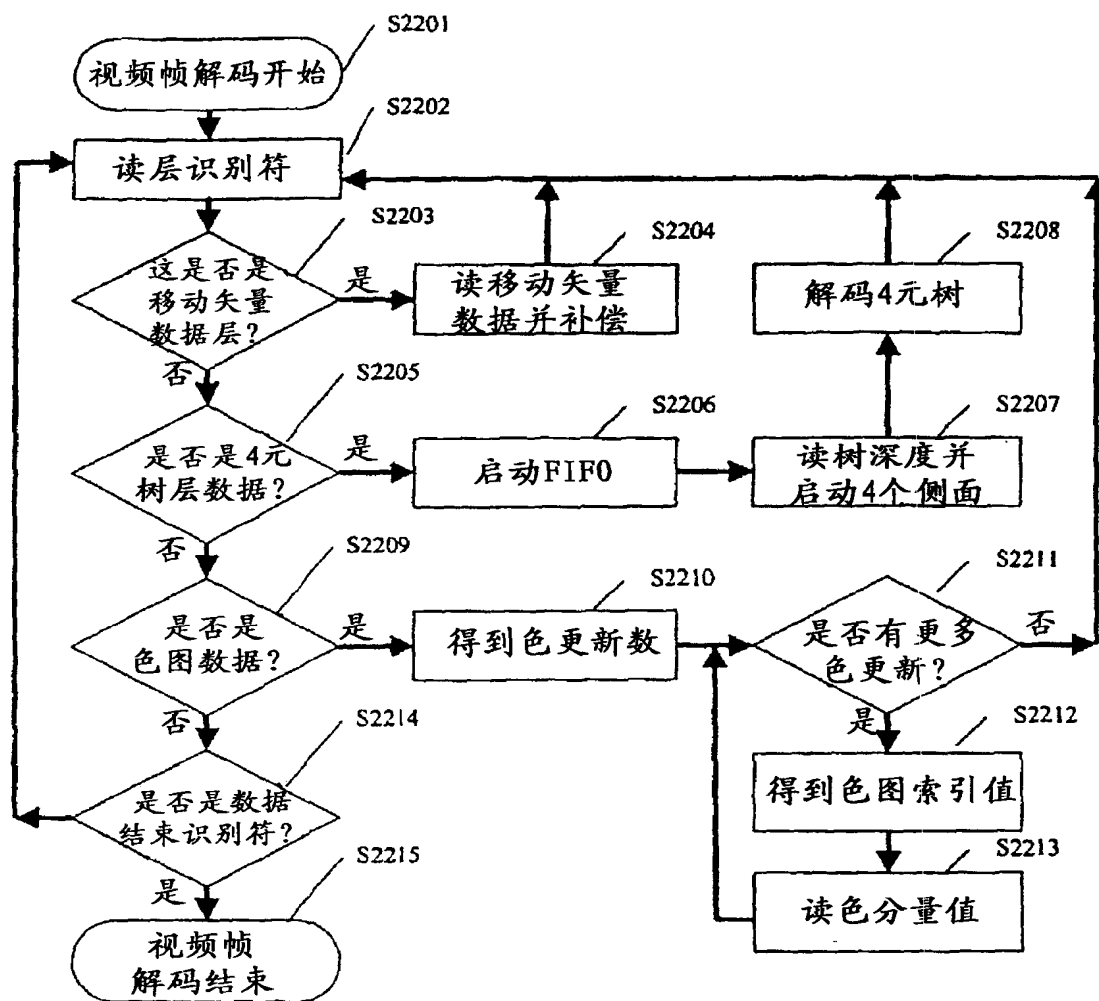


图 49

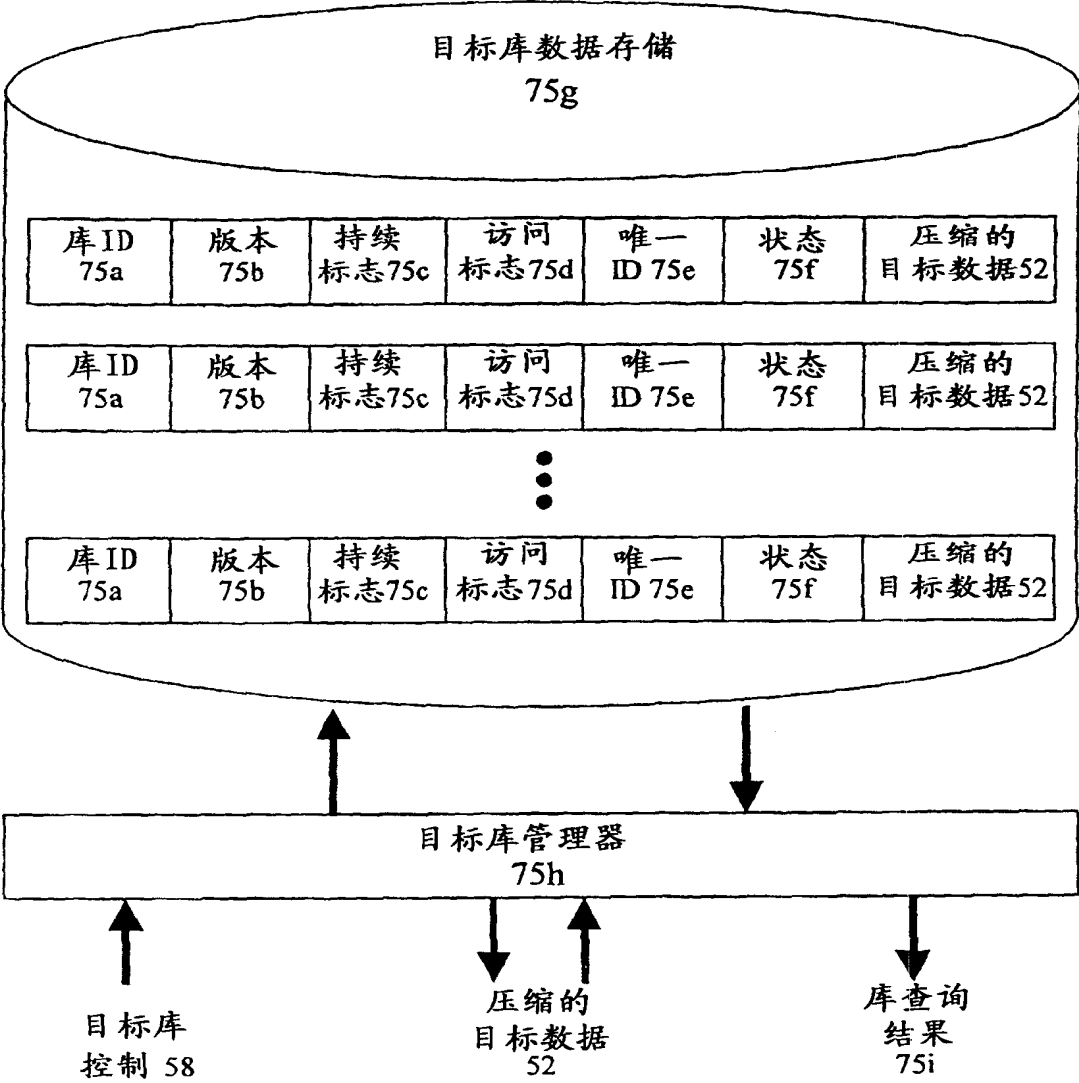


图 48

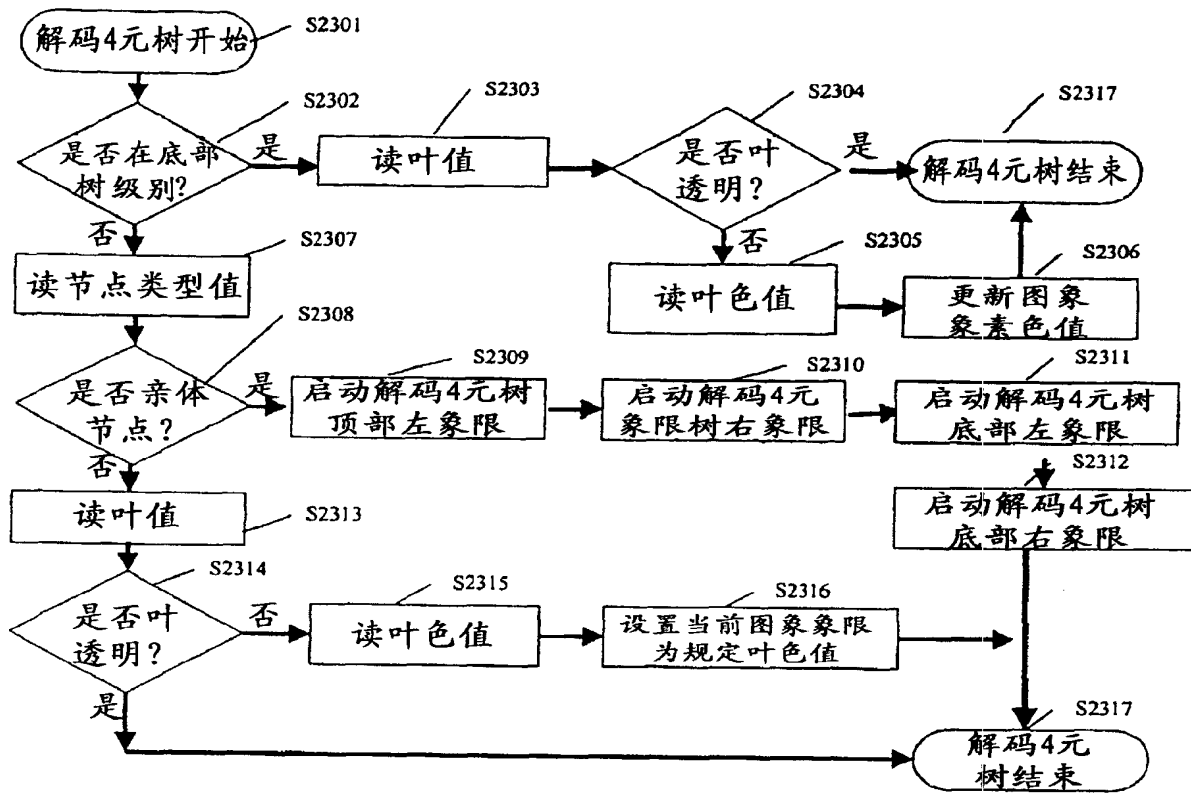


图 50

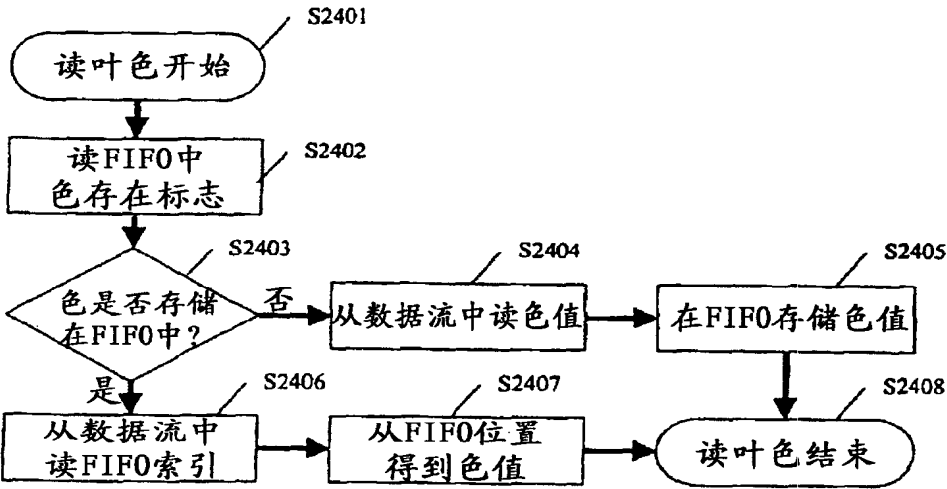


图 51